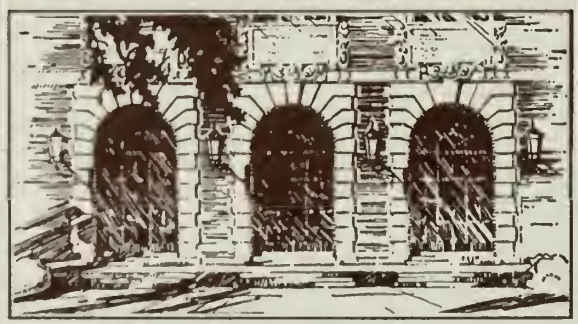
LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84
Il6r
no. 361-366
cop. 2



The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

To renew call Telephone Center, 333-8400

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

MAR 5 1980

MAR 8 REC'D

L161—O-1096

math

Report No. 365

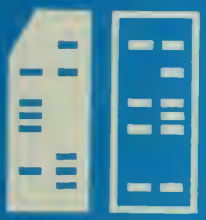
WEATHER ANALYSIS ON A PARALLEL COMPUTER

by

David Martin Goddard

December 1, 1969

THE LIBRARY OF THE
OCT 3 1969
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

Report No. 365

WEATHER ANALYSIS ON A PARALLEL COMPUTER*

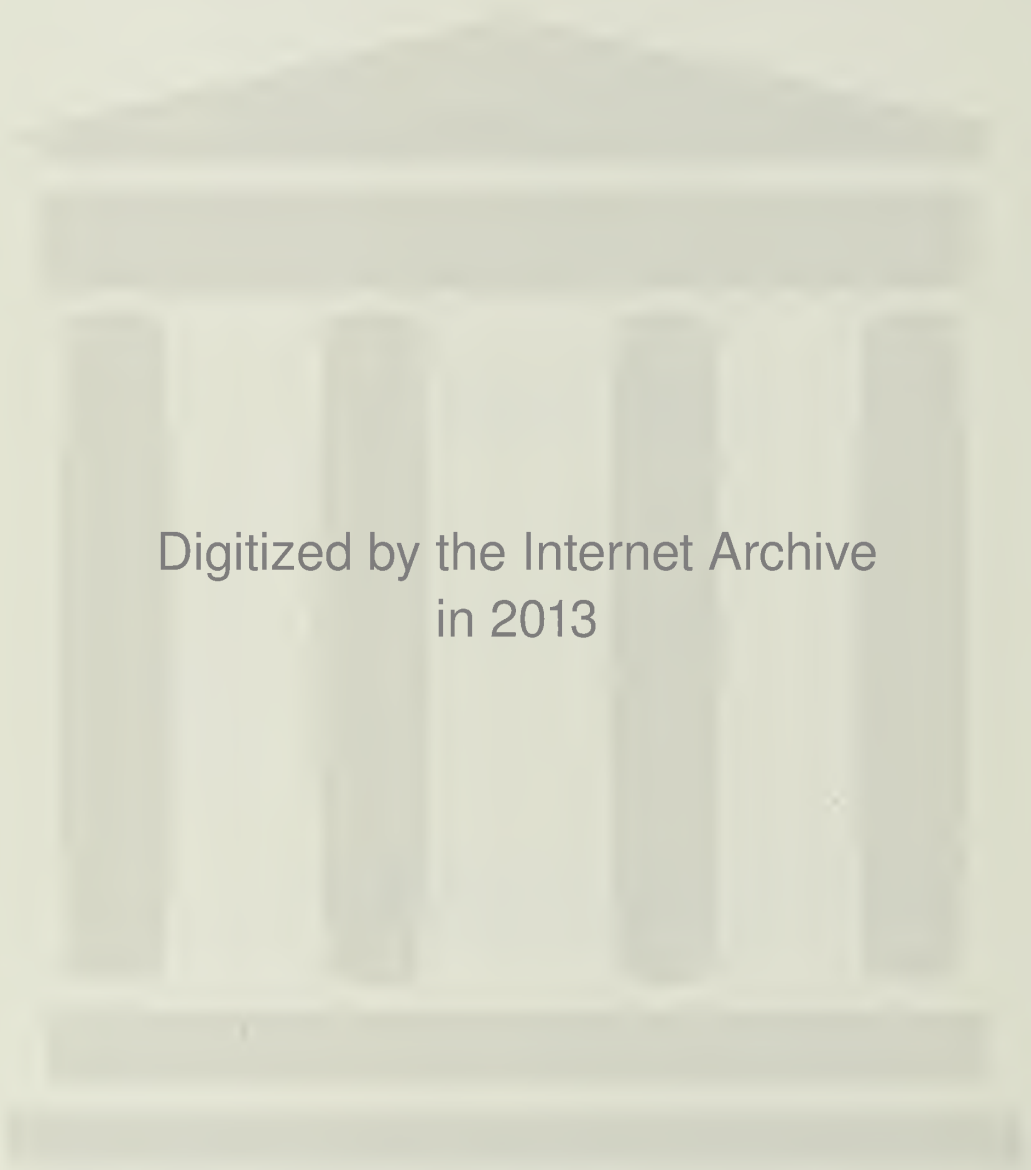
by

David Martin Goddard

December 1, 1969

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

*This work was supported in part by the Advanced Research Projects Agency as administered by the Rome Air Development Center under Contract No. AF 30(602)-4144 and submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, September, 1969. The views expressed herein are those of the author and do not necessarily reflect those of the United States Air Force.


Digitized by the Internet Archive
in 2013

http://archive.org/details/weatheranalysiso365godd

ABSTRACT

The numerical weather analysis procedures of the National Meteorological Center are reviewed, and problems in adapting them to a parallel computer are considered. Algorithms for storage allocation, processing teletype data, dynamically arranging data, and producing the analyzed field are presented.

ACKNOWLEDGMENT

The author appreciates the interest and encouragement of his thesis advisor, Professor David J. Kuck. The information provided by Mr. James McDonell of the National Meteorological Center was most helpful. Major Harold D. Woods of the Forecaster Training Branch, Chanute Technical Training Center, and the personnel of the base weather station at Chanute Air Force Base were helpful in providing reference material.

Also, the author extends his appreciation to the B5500 operating staff and key punch operators. The assistance of Mr. James Madden of the ILLIAC IV professional staff is very much appreciated. And, of course, without the typing abilities of Mrs. Patricia Douglas this final form would not be.

The Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, sponsored the author's studies at the University of Illinois.

TABLE OF CONTENTS

Page

1. INTRODUCTION 1

2. NUMERICAL WEATHER ANALYSIS 2

2.1 Relation to Numerical Forecasting 2

2.2 Dynamics and Thermodynamics for Upper Air Analysis 2

2.3 The National Meteorological Center (NMC) Procedures 5

2.4 The NMC Grid 10

3. THE ILLIAC IV COMPUTER 12

3.1 Organization 12

3.2 Control Unit 12

3.3 Processing Elements 14

3.4 Memory Addressing 14

3.5 Mode Control 17

3.6 Routing 17

3.7 Input/Output and Array Control 18

4. STORAGE ALLOCATION 19

4.1 Data Organization and Arithmetic Efficiency 19

4.2 Arrangement of Grid Points 20

4.3 Neighboring Points and Responsibility Areas 21

5. PROCESSING TELETYPE REPORTS 25

5.1 Structure of Teletype Messages 25

5.2 Scanning and Recognition 25

Page

5.3 Decoding 29

5.4 Hydrostatic Checking 30

5.5 Error Recovery 30

6. RELOCATION AND ANALYSIS OF DATA 31

6.1 Initial Assignment of Stations to PEs 31

6.2 Interpolated Values and Consistency Check 31

6.3 Interchange of Field Values 32

6.4 Analysis Calculations 33

6.5 Analyst Interaction 35

7. CONCLUSIONS 36

LIST OF REFERENCES 37

APPENDIX 38

LIST OF FIGURES

Figure		Page
1.	The NMC Analysis Grid	11
2.	Control Unit	13
3.	Processing Element	15
4.	Data Word Formats	16
5.	Assignment of Grid Points to PEs	22
6.	Arrangement of Points within PE	23
7.	Sample Teletype Data	26
8.	Data Organization During Analysis	34

LIST OF TABLES

TABLE Page

1. EXAMPLE OF RADIUS OF INFLUENCE AS A FUNCTION OF DATA DENSITY . . . 7

2. ORGANIZATION OF DATA FOR ANALYSIS 20

1. INTRODUCTION

Numerical weather analysis was developed to provide improved input to numerical forecasting programs. Meteorological data is obtained by observation from stations that are located neither in a regular pattern nor uniform distribution. This data, which is transmitted via teletype in standard code formats, must be collected, decoded, and reduced to the grid used by the forecasting routine. The National Meteorological Center has developed effective procedures that produce operational analysis charts, as well as input to forecasting programs.

The ILLIAC IV computer will be an array of arithmetic units operating together under central control. In a one quadrant array, sixty-four processing elements can each be doing two 32-bit arithmetic operations simultaneously.

Implementation of a numerical weather analysis program on ILLIAC IV requires consideration of operation on teletype reports in some parallel manner, storage of data so that it is readily accessible to appropriate processing elements, and maximization of simultaneous operations across the grid.

2. NUMERICAL WEATHER ANALYSIS

2.1 Relation to Numerical Forecasting

From the time von Neuman and Charney investigated automated forecast procedures, meteorologists realized that input data was a major problem. Indeed, part of Richardson's difficulties in his early attempt at manual numerical integration to produce prognostic information was the lack of upper air data* [1]. It is the atmospheric layers above the surface to which dynamic and thermodynamic equations have been most successfully applied; upper air observing stations are considerably fewer in number than surface ones.

Manual analysis is highly subjective and time consuming. The same objections apply to manual extraction of data from charts for input to machine analysis. In sharp contrast, the operation of a numerical forecast program is quite fast and as objective as the atmospheric model it simulates.

2.2 Dynamics and Thermodynamics for Upper Air Analysis

The equations of motion applicable to a parcel of air on the rotating earth can be simplified by assuming that the only forces present are those due to pressure gradient, gravity, and the earth's rotation, and that motion reaches an equilibrium state [2]. The simplified equations are

$$fv = \frac{1}{\rho}\frac{\partial p}{\partial x} \qquad fu = -\frac{1}{\rho}\frac{\partial p}{\partial y} \qquad g = -\frac{1}{\rho}\frac{\partial p}{\partial z} + 2\Omega \cos\varphi$$

where x is positive in the eastward direction, y northward, and z upward; u and

---

*Additionally, Richardson's work failed because the compensation of horizontal forces was not considered and because of computational instability in his model.

v are the x and y components of velocity, respectively; p is pressure, ρ is density, g is acceleration of gravity, and $f = 2\Omega \sin\varphi$ is the Coriolis parameter at latitude φ for the earth's angular velocity Ω. The third equation is the equation for hydrostatic equilibrium* after the small vertical Coriolis term is neglected. The horizontal equations are termed the geostrophic wind equations and express the horizontal balance of Coriolis and pressure gradient forces.

When the vertical coordinate is transformed to pressure, the horizontal equations are

$$v = \frac{g}{f}\left(\frac{\partial z}{\partial x}\right)_p \qquad \text{and} \qquad u = -\frac{g}{f}\left(\frac{\partial z}{\partial y}\right)_p$$

In analysis work these equations have the advantage that density is not explicitly involved. The constant pressure surfaces are sufficiently quasi-horizontal that they are easily usable in place of constant height surfaces.

The hydrostatic equation, ignoring horizontal forces, is

$$dp = -\rho g\, dz.$$

Substituting the ideal gas equation of state and integrating from z_1 to z_2 yields [3] the hypsometric equation

$$z_2 - z_1 = \frac{R\bar{T}^*}{mg}(\ln p_1 - \ln p_2),$$

where m is the average molecular weight of dry air, R is the gas constant, and

---

*Hydrostatic equilibrium is one of the most basic assumptions made in theoretical meteorology. It quantifies the fact that the vertical scale of motion is orders of magnitude smaller than the horizontal one.

$\overline{T}^{*}$ is the mean virtual temperature of the layer. Virtual temperature, T^{*}, is the temperature of dry air of the same density and pressure as the given moist sample. It rarely exceeds the actual temperature by more than three degrees Celcius.

Thus the thickness of a layer of air between two constant pressure surfaces is directly proportional to the mean virtual temperature of the layer. From a plot of temperature and moisture content versus logarithm of pressure, the virtual temperature plot can be easily developed. Equal area averaging of the virtual temperature between two pressures then yields the thickness of the layer. Because of the averaging nature of the hypsometric equation, relatively simple upper air sounding equipment will yield good measurements of the heights of constant pressure surfaces.

The hydrostatic, geostrophic, and hypsometric equations contain the factor g, the earth's acceleration of gravity. Meteorologists usually combine the acceleration due to gravitational forces with the centrifugal force of the earth's rotation into g. Thus g varies slightly with latitude and surfaces of equal geopotential (specific potential energy due to gravity) do not correspond to those of geometric altitude. The above equations are usually formulated in terms of geopotential rather than geometric height. The geopotential meter is defined from

$$\frac{1}{9.8}\int_0^z g \, dz$$

and gives geopotential height measurements numerically similar to geometric height in meters [2]. Actually, geopotential meters are a measure of specific energy.[**]

[**] In much of the development of theoretical meteorology, specific volume, specific force, etc., are used and explicit measure of the mass of air and air parcels is not required.

In summary, the hydrostatic and geostrophic equations express the equilibrium that is the general nature of the atmosphere. The geostropic equations yield a simple relationship between wind velocity and the gradient of pressure, or equivalently, of geopotential height. The hypsometric equation provides a simple means of determining geopotential height of constant pressure surfaces.

2.3 The National Meteorological Center (NMC) Procedures

Cressman [4], later at the Joint Numerical Weather Prediction Unit (the organizational predecessor of NMC), considered an analysis procedure for geopotential height based on quadratic least squares interpolation of observed data at grid points. The asymmetry of data distribution, particularly over ocean areas, caused poor results in such areas of sparse data. Furthermore, extensive computation was required to invert the matrices of coefficients during the computation at each grid point.

Bergthorsson and Doos [5] developed a procedure at the University of Stockholm based on modification of a first-guess field. Their reasoning was that interpolation of any nature will fail in an area lacking observational data and that in such areas a forecast based on old data is better than an interpolated analysis. In tropical areas, climatological normals are relatively good first guesses.

Their procedure begins with determination of a first-guess field as a weighted average of the climatological normals and the numerical forecast based on the data from the preceding observation time. Next, three types of corrections to the first-guess field are determined for each reporting station. The first correction assumes that the difference at the station of observed height (z_0) and first-guess height (z_{fs}, as determined by interpolation) is equal to the difference at surrounding grid points of the actual height (z_1)

and first-guess height (Z_f)

$$z_1 = z_f + (z_0 - z_{fs}).$$

By assuming the observed wind is geostrophic and representative of the interval between the station and the point, a second corrected value (z_2) is obtained

$$z_2 = z_0 + \left(\frac{\partial z}{\partial x}\right)_0 \Delta x + \left(\frac{\partial x}{\partial y}\right)_0 \Delta y$$

where

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2}$$

is the distance in grid units from station to grid point and $\left(\frac{\partial z}{\partial x}\right)_0$ and $\left(\frac{\partial z}{\partial y}\right)_0$ are the x and y components of height gradient as computed from the observed wind.

The third corrected value is similar to the second but uses the gradient at the grid point. The relative weighting (w_1, w_2, w_3) of the three corrected values was determined by statistical regression. For stations reporting only height, the correction z_1 is computed but a different weighting function is used.

Because of the correlation of data from stations in areas of high station density, the composite corrected value is further weighted by the inverse of station density. The computed value at each grid point is then the weighted average of all composite corrections for stations and of the first-guess value. One scan of the grid was used by Bergthorsson and Doos in most of their experimental work.

Cressman [6] modified the Bergthorsson and Doos procedure by adopting a simplified weighting function and making multiple scans of the grid (see Figure 1 for NMC grid). This modification also drops the third type of correction. The weighting function, the same for both types of corrections, is

$$w = \frac{R^2 - d^2}{R^2 + d^2}$$

where R, the radius of influence, is the maximum d for which a station is used to correct a grid point. This type of weighting function effects a smoothing of all data within distance R. The value of R is decreased on each of four consecutive scans in order that the phenomenon of decreasing scale can be included in the resulting final analysis while errors in the first-guess field are quickly smoothed out.

In order that height data may be given greater weight, wind data (correction 2) is not included on the first scan of a height analysis. Stations reporting wind only are utilized by forming a type 2 correction (z_2') with z_0 replaced by z_{fs}.

Since 1964, the value R (the radius of influence) at a grid point is determined by data density [7]. This variation of R is a discrete function of the number of height reports (see Table 1).

TABLE 1

EXAMPLE OF RADIUS OF INFLUENCE
AS A FUNCTION OF DATA DENSITY

R in grid units	Number of height reports
2.2	5 or more
3.5	7 or more
6.9	1 or more

The calculated value at the grid point is

$$z = z_f + \frac{A\sum_{i=1}^{\ell} w_i c_i + B\sum_{j=1}^{m} w_j c_j + \sum_{k=1}^{n} w_k c_k}{A\ell + Bm + n}$$

$$c_i = (z_1 - z_f)_i = z_0 - z_{fs}$$

$$c_j = (z_2' - z_f)_j = z_{fs} + \left(\frac{\partial z}{\partial x}\right)_i \Delta x_j + \left(\frac{\partial z'}{\partial y}\right)_0 \Delta y_j - z_f$$

$$c_k = (z_2 - z_f)_k = z_k + \left(\frac{\partial z}{\partial x}\right)_k \Delta x_k + \left(\frac{\partial z}{\partial y}\right) \Delta y_k - z_f$$

The subscripts i, j, and k indicate values for one of the $\ell + m + n$ stations that can influence the report. The coefficient A can be regarded as the weight given to gradients of the first-guess field in relation to the observed winds, weighted as 1 [6]. A equals $\frac{1}{8}$. B similarly is the weight given to the interpolated first-guess value as a basis for extrapolating the gradient [7]. B = 0, $\frac{1}{8}$, $\frac{1}{4}$, and $\frac{1}{2}$ on scans 1, 2, 3, and 4, respectively.

The value of the components of height gradient $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$ are computed relative to the grid coordinates. If the geostrophic gradients are determined by

$$\left(\frac{\partial z}{\partial x}\right)_g = \frac{f}{g} v \qquad \left(\frac{\partial z}{\partial y}\right)_g = \frac{-f}{g} u$$

then an approximation of non-geostrophic affects can be included by computing

$$\frac{\partial z}{\partial x} = \left(\frac{\partial z}{\partial x}\right)_g + \frac{gm^2K}{f^2\delta^2}\left(-\left(\frac{\partial z}{\partial y}\right)_g D'_{xy} + \left(\frac{\partial z}{\partial x}\right)_g D'_{yy}\right)$$

$$\frac{\partial z}{\partial y} = \left(\frac{\partial z}{\partial y}\right)_g + \frac{gm^2K}{f^2\delta^2}\left(\left(\frac{\partial z}{\partial x}\right)_g D'_{xx} - \left(\frac{\partial z}{\partial y}\right)_g D'_{xy}\right).$$

Again f and g are the Coriolis parameter and acceleration of gravity, respectively; m and δ relate to the map on which the grid is based. K is a constant less than one which is assigned to a constant pressure surface for a season. D'_{xx}, D'_{yy}, and D'_{xy} are the second derivatives of the height field. On the square grid used by NMC, the nearest eight points to a given grid point are required to compute these derivatives.

Before each scan of the grid, each station report is checked for its consistency with its neighbors. The mean difference of each station within two grid units from the first-guess field is compared with the difference for the station under consideration. The station's report is marked for acceptance or rejection on the basis of this comparison and data density.

The source first-guess fields used by NMC vary. The 1000 mb, 500 mb, and 200 mb first-guess fields for height are obtained by adjusting in the tropical regions the preceding forecast by climatological height gradients. For some levels, vertical extrapolation or interpolation from levels previously analyzed provides the first-guess field.

Except for the 500 mb level, temperature is analyzed at the same time as height. The temperature analysis is like that above but omits gradients. The u and v components of wind are analyzed by taking the geostrophic winds from the corresponding height analysis as the first-guess. Two scans are used to adjust the field to reported data. Other modifications to the basic procedure are used to accomplish objective tropopause analysis and wind fields in the tropics [8]. The latter uses a grid on a Mercator map projection.

2.4 The NMC Grid

The grid used by NMC (Figure 1) for both analysis and forecast procedures is obtained by drawing one-half inch squares on a 1:30,000,000 scale polar stereographic projection of the northern hemisphere. This projection is true at 60° N latitude. The resulting grid unit δ is 381 kilometers or 206 nautical miles. The map factor, the ratio of projection image distance to distance on the earth, is

$$m = \frac{1 + \sin 60°}{1 + \sin \varphi} .$$

The grid is oriented so that the meridians 10° E, 80° W, 170° W, 100° E are along grid lines. The grid is numbered so that the i (x coordinate) and j (y coordinate) zero lines intersect in the Indian Ocean and the North Pole is i = 33, j = 26. The following formulas may be used to convert the latitude and longitude of any location to grid coordinates

$$r = \frac{R_e \cos \varphi}{1 + \sin \varphi}$$

$$i = ip + r \sin(\Psi + 100)$$

$$j = jp + r \sin(\Psi + 100)$$

$$R_e = 21.204 \text{ grid units}^4$$

where ip and jp are the grid coordinates of the pole and Ψ is west longitude.

The NMC analysis procedures use the 1977 points enclosed by the octogon depicted in Figure 1.

---

[4] R_e is the radius of the earth on the map projection image plane--R_e = 6371.2 km (1 + cos 30°)/381 km per grid unit.

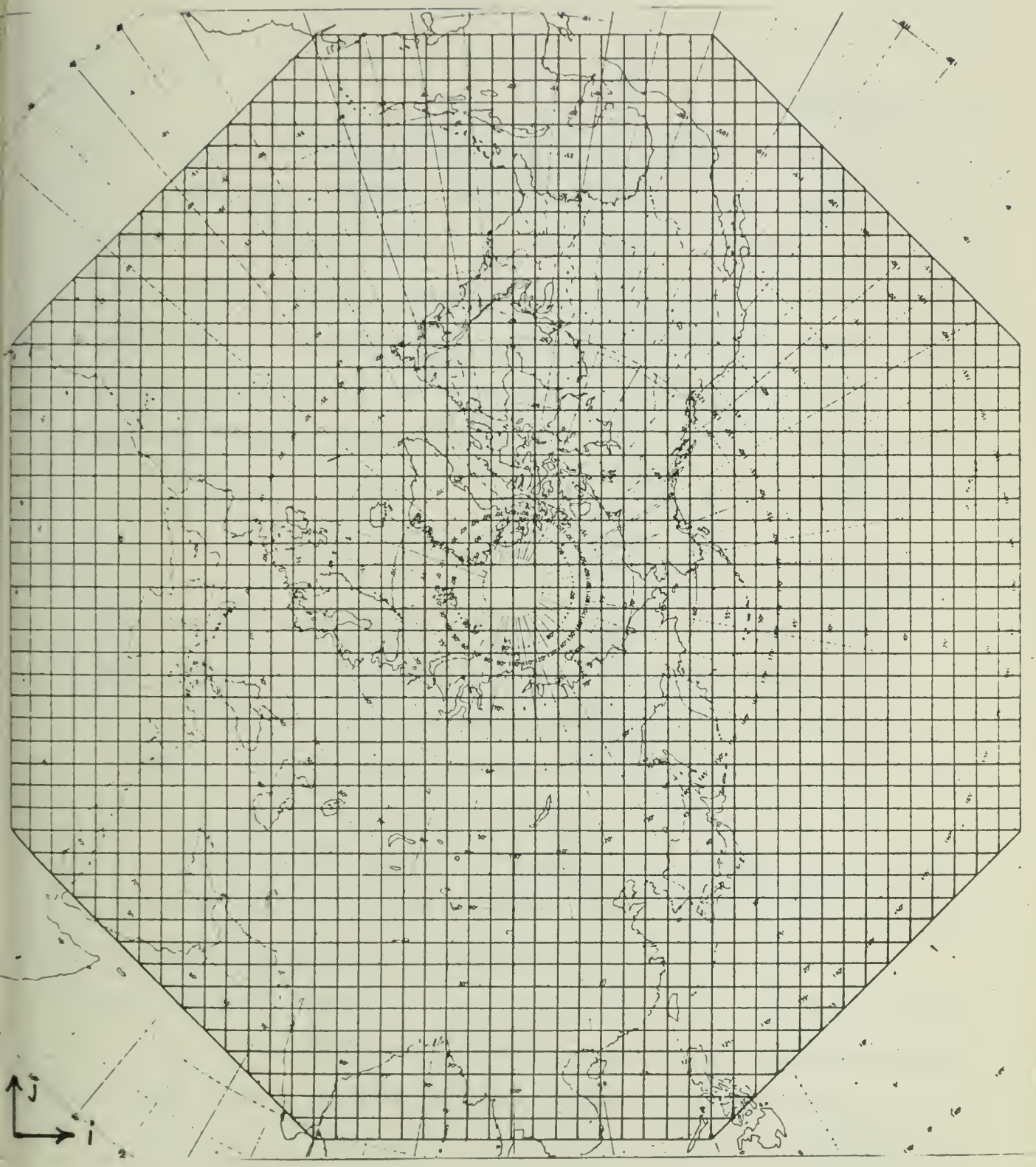


Figure 1. The NMC Analysis Grid.

3. THE ILLIAC IV COMPUTER

3.1 Organization

The basic concept of ILLIAC IV is that the rate of calculation on a digital computer can be increased by executing more than one arithmetic operation simultaneously. Furthermore, the parallel execution of arithmetic operations does not require duplication of control functions. To this end, ILLIAC IV has 64 arithmetic units, or processing elements (PEs), under the control of one control unit (CU). The combination of one CU and 64 PEs is termed a quadrant and one, two, or four quadrants can be united to operate as an array [9].

3.2 Control Unit

The control unit (Figure 2) fetches and decodes all instructions. Only address arithmetic (24-bit, fixed point) can be done in the CU. Instructions that only involve the CU include branching instructions that allow the entire quadrant to execute from another part of the instruction stream.

The advanced station (ADVAST) in the CU does the initial decoding of instructions. It contains four accumulator registers (ACARs) and 64 words of data buffer (ADB). Transfer instructions are entirely executed by ADVAST.

The final station (FINST) completes the decoding of all instructions requiring PE action. This includes data fetches to ADVAST. A queue of up to eight instructions is maintained between ADVAST and FINST.

The instruction look-ahead unit (ILA) maintains a buffer of up to 128 instructions and fetches all instructions from memory.

The test and maintenance unit (TMU) is the only path to or from outside control. Loading TMU registers can affect initialization of the control unit.

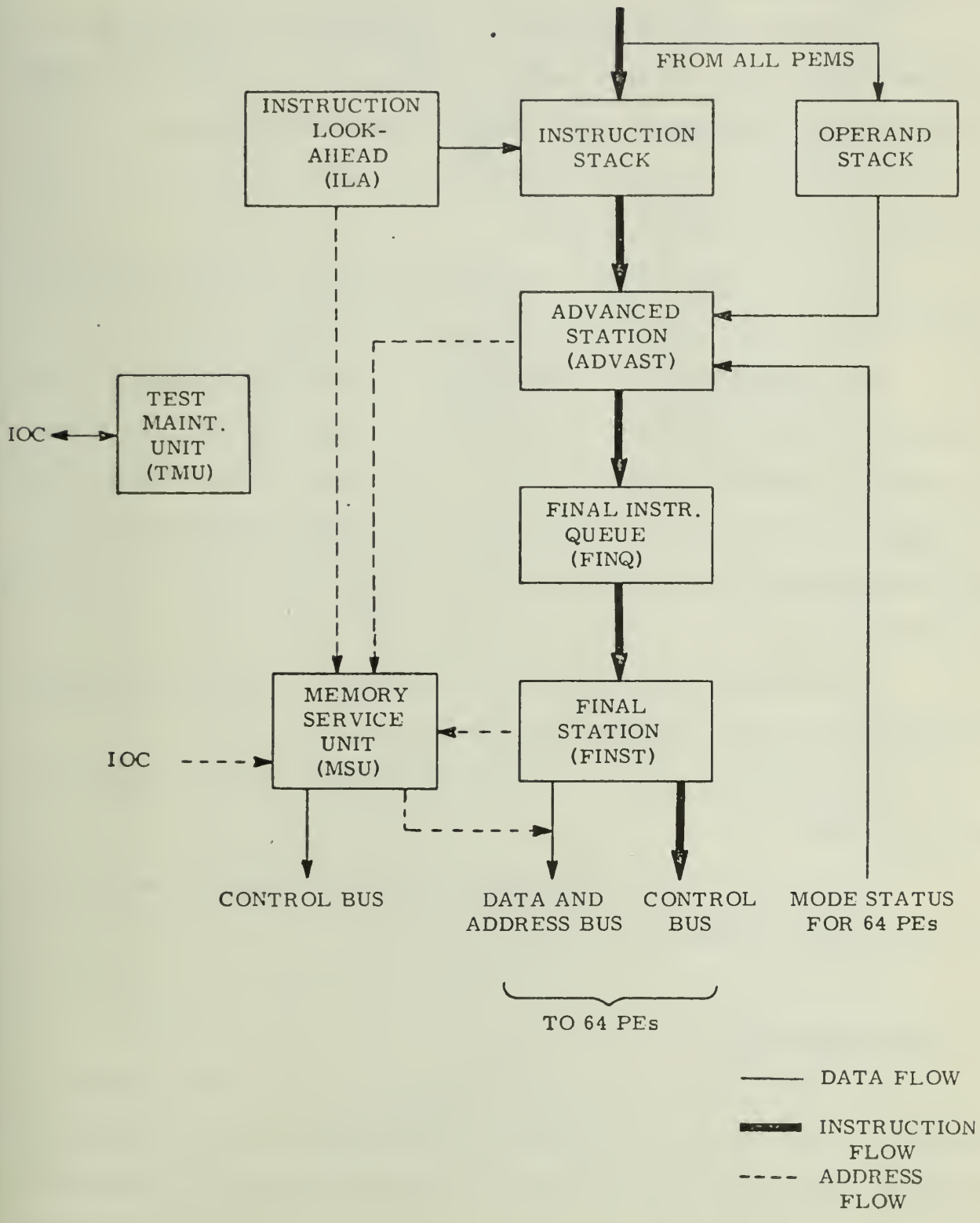


Figure 2. Control Unit.

3.3 Processing Elements

Processing elements (PEs, Figure 3) execute arithmetic, boolean, shift, and data transfer instructions in lock-step under sequencing from FINST. Arithmetic instructions can be either 64- or 32-bit floating point or 48- or 24-bit fixed point (see Figure 4). There are also a limited number of instructions for operation on eight-bit bytes.

Each processing element is assigned 2048 words of storage (PEM). The term processing unit (PU) denotes the PE plus its storage.

The addressable PE registers are the A (accumulator) register, the R (routing) register, the D (mode) register, the X (index) register, the B register, and the S register. Register C (the carry register in the arithmetic logic) is addressable in connection with byte operations by only one instruction. Register X is 16 bits in length; register D is eight bits (see 3.5). All others are 64 bits long.

The S register can be used to store operands. Its 16 low order bits can function the same as register X for index operations.

The 64-bit wide common data bus (CDB) allows the CU to broadcast operands to all PEs.

Register X and register S indexing may be done on shift operations as well as on memory fetches and stores.

3.4 Memory Addressing

The control unit addresses memory such that the low order six bits indicate a particular PE. The preceding 16 bits address a row of 64 words in PE memory (five of these allow for expansion from the present $2^{11} = 2048$ words per PE). Indexing by an ACAR is by word; by register X or S, by row (low order bits discarded).

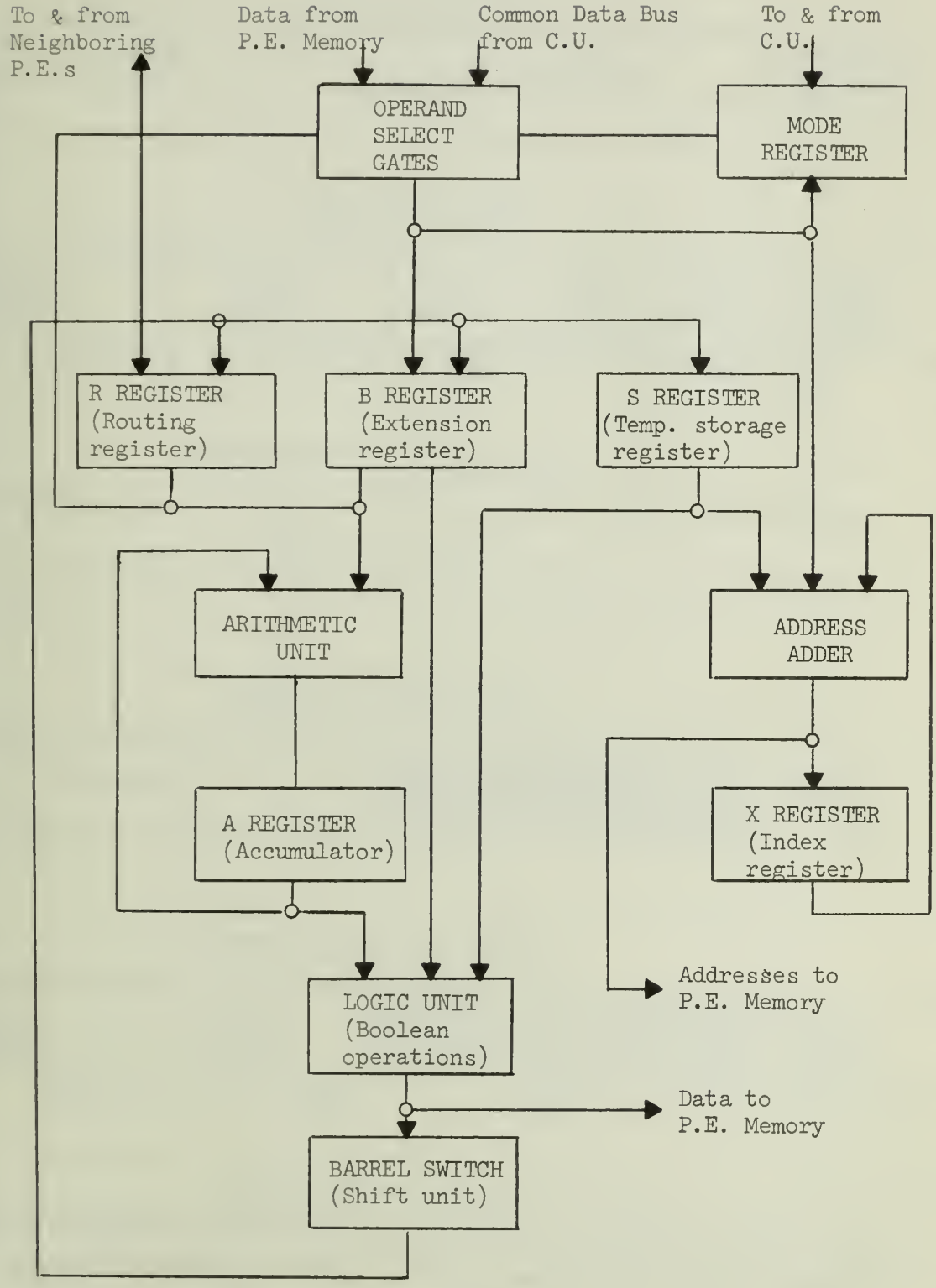


Figure 3. Processing Element.

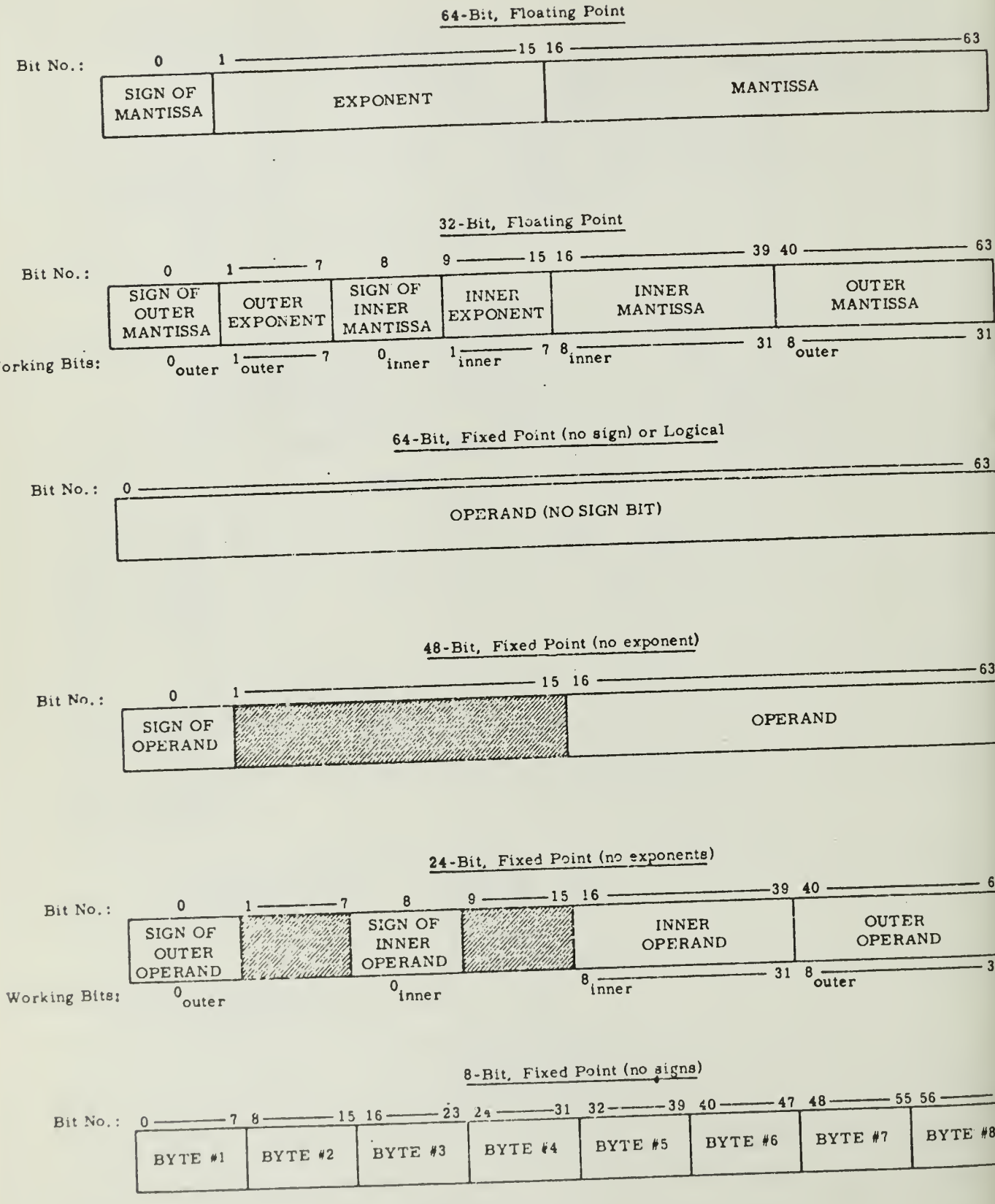


Figure 4. Data Word Formats.

3.5 Mode Control

Input to storage and to three of the programmer addressable PE registers (A, S, and X) can be inhibited by mode setting operations. Thus during instruction execution, PEs can be "turned off", and different PEs can in effect execute different instruction streams at different times. Note, however, that even if all PEs are disabled, FINST sequences through the instructions. The first two bits of register D, the E and E1 bits, are the enable bits.

A test on register A or an index test on register X (or the low 16 bits of register S) results in setting the I or J mode bits (in 32-bit operations, the G or H bits are also set by the results of the tests on the inner word of register A).

The F and F1 bits reflect faults (exponent overflow, for example). The setting of an F bit results in an interrupt in the CU.

The SET instruction is available to set any mode bit from the results of a logical AND or of an inclusive OR of any mode bit and either the E or the E1 bit. For example, SET I J AND -E1 will set bit I to the result of J and not E1. Mode bits may also be set by broadcast from the CU. In this situation, each bit of an ACAR goes to the designated bit in the PE of corresponding number (i.e., bit 0 to PE 0, bit 1 to PE 1, etc.).

3.6 Routing

Because a given memory location is directly accessible to only one PE, routing operations are implemented to allow transmissions of data from the designated register to register R of another PE. A route of length one or eight takes one clock cycle [10]. Routes of other distances are accomplished by a sequence of routes of distance eight and one.

3.7 Input/Output and Array Control

The only means of input/output for ILLIAC IV is through a disk system with a transfer rate of 0.5 billion bits per second and maximum access time of 40 milliseconds [9]. A Burroughs 6500 computer channels data to and from the disk system.

The B6500 also has executive control of the array. Through the TMU it can initiate a program, accept I/O requests from the operating program, return results of I/O operations, and halt the array.

4. STORAGE ALLOCATION

4.1 Data Organization and Arithmetic Efficiency

In order that the analysis of a field on the NMC grid may be executed most efficiently, it is necessary that the number of operations executed in each PE be as nearly equal as possible. The analysis routine is based on adjusting the first-guess field at each grid point. Therefore, allocation of grid points equally is a first step. Unfortunately, reported data has a density far from uniform. Furthermore, there are reports, such as aircraft observations, that will not occur in the same location with each analysis. Thus, some execution time procedure must be devised to associate the reported data and the grid points.

Use of 32-bit fixed point (24-bit mantissa only) arithmetic will maximize arithmetic efficiency while maintaining accuracy consistent with observational accuracy. Furthermore, 16 bits of accuracy in the observed quantities is sufficient if they are translated against some fixed value. For instance, 500 milibar heights expressed in whole meters with a base of 3500 geopotential meters allows computation with 11 bit quantities[1]; 3500 to 7595 is sufficient range for this analysis. Positive translation against some lower bound is preferable to positive or negative differences about a mean or standard value in that the latter would require programmatic setting of the sign bit after unpacking the data. The organization of 16 bit quantities in a block of four 32-bit words is shown in Table 2.

---

[1] Multiplication of two 11-bit fixed point quantities results in 22 bits of product and is within the 24 bits allowed.

TABLE 2

ORGANIZATION OF DATA FOR ANALYSIS

Word number	Contents of Upper Half	Contents of Lower Half
0	i coordinate	j coordinate
1	average temperature	average height
2	observed temperature	observed height
3	i wind component	j wind component

4.2 Arrangement of Grid Points

The assignment of grid points to PEs is governed by two somewhat conflicting requirements. With 1977 points to be assigned to 128 PEs*, with four quantities (height, temperature, and two components of height gradients as computed from observed winds) at ten standard constant pressure levels required by the analysis system, one report may present forty quantities to be used in various analyses. If only 200 of the better than 1000 reports per analysis contain all 40 data items, there would be far more data items than can be contained in one PE. The total report must then be kept in one location only. Since a complete analysis at one level of both temperature and height can be accomplished with only eight pieces of data**, this data can be extracted from each report and associated with all appropriate grid points. Blocks of 15 to 16 points must be arranged so that on one hand the total number of stations to be used by the points is evenly distributed, and on the other hand the total

---

*Actually "half" PEs, the inner/outer portions of the arithmetic units will be termed "PEs" as their functions during the analysis procedure are essentially independent. The inner words will be referred to with even numbers; the outer, with odd.

**In addition to the four observed quantities, station location and the first-guess height and temperature values (as interpolated from surrounding grid points) are required.

number of stations assigned to each PE is kept to a reasonable number. A totally random distribution of the points for a PE across the hemisphere would be disastrous. With each point possibly being influenced during analysis by stations up to 6.9 grid units away [7] such a distribution would force each PE to store practically every station. An arrangement of four by four blocks, however, would result in blocks over western and central Europe with every point being influenced by many stations.

Figure 5 indicates an empiracally determined arrangement that not only balances the above two objectives but also results in a simple algorithm to interchange grid point values for neighboring points. The blocks are eight points long in the increasing i direction, two wide in the increasing j direction. The points indicated by **o** in Figure 5 are those of smallest i and j values for each group of 16.

Coverage of the octagon used by NMC is not complete; 40 points at boundaries are omitted. 19 of these points are the j = 51 row and only one (i = 49, j = 10) is as distant from the octagon boundary as the diagonal of a grid square. Only two rows of three blocks each do not start in columns i = 9, 17, 25, 33, 41, or 49, and these six are exactly four points "out of phase".

4.3 Neighboring Points and Responsibility Areas

With the assignment of points for which analysis computations will be done, the problems of determining average (interpolated) field values and non-geostrophic height gradients for each station can be solved. In Figure 6, the encircled points are those to be assigned to a PE for the analysis calculation. The inner rectangle, three by nine grid squares in size, covers the

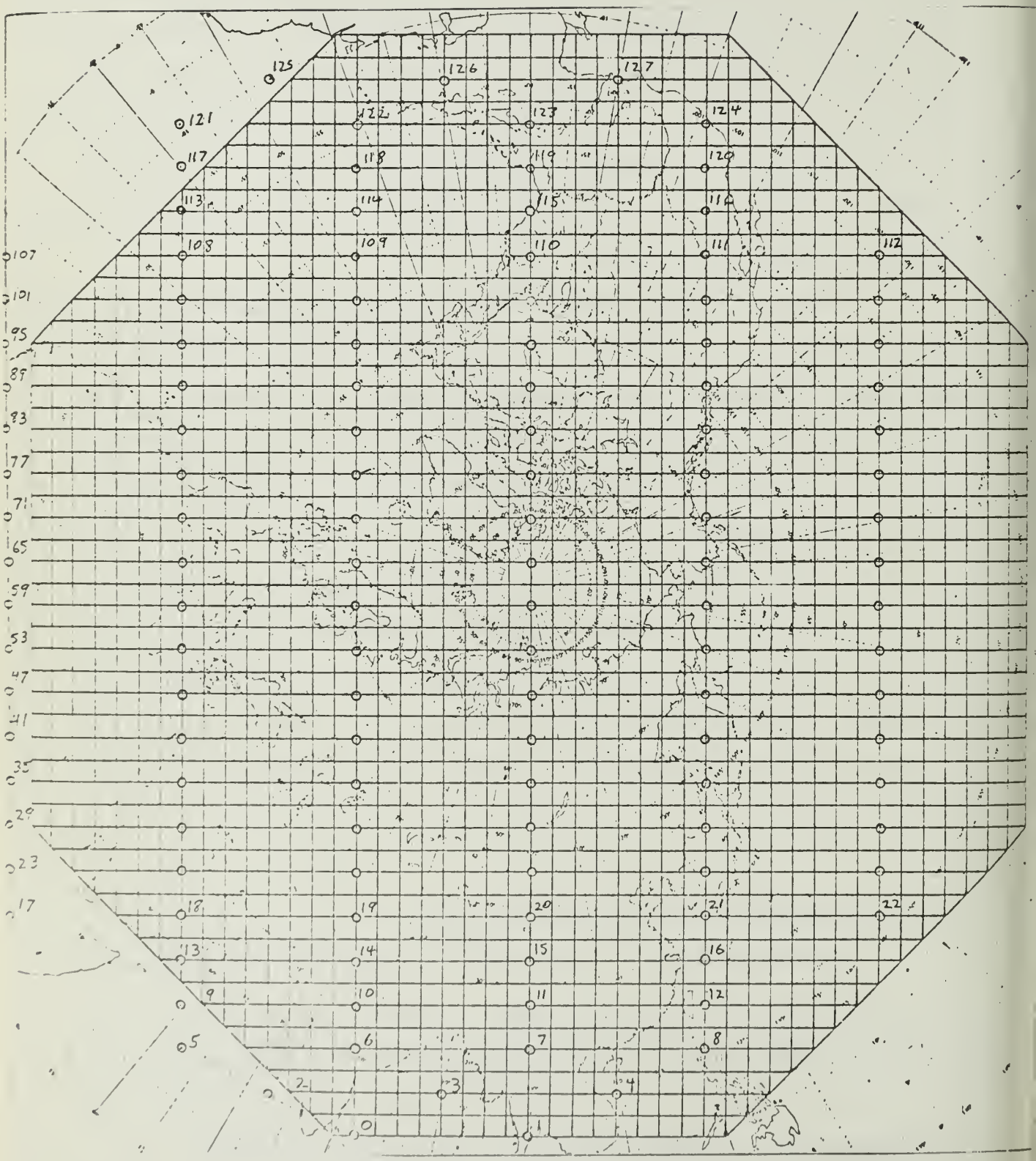


Figure 5. Assignment of Grid Points to PEs.

Figure 6. Arrangement of Points within PE.

responsibility area for that PE*. The stations falling within this area will be the responsibility of the PE for calculation of average values and height gradients and for consistency check. An array of length 91 in each PE provides analysis field data for these requirements.

The consistency check of observed data against first-guess field values requires inclusion of the outer rectangle-points 0 to 13, 25, 26, 38, 39, 51, 52, 64, 65, and 78 to 90. The inclusion of these points allows interpolated field values to be found for any station within two grid units of the responsibility area.

---

*A station falling on a boundary would be assigned to the higher number PE.

5. PROCESSING TELETYPE REPORTS

5.1 Structure of Teletype Messages

Figure 7 shows a portion of a teletype message that contains two upper air reports--those from Topeka (TOP) and Dodge City, Kansas, for 1200 Greenwich Mean Time (GMT), May 8, 1969. Appendix A gives a grammar for this type of report, the type transmitted by North American stations [11], in BNF meta-language. The syntactic structure is relatively simple. Thus it is possible to apply the methods of Heimerdinger [12] to scan and parse this language of teletype reports.

Unfortunately, however, this is not the only form for messages transmitted over the national and international meteorological circuits. Because of the slow rate of teletype transmission, it would also be undesirable to tie up ILLIAC IV with the task of scanning incoming teletype messages. It is therefore appropriate to use the B6500 computer to receive the messages. While sorting out types of messages, the B6500 can remove the communications system portions of the incoming stream. The first three lines in Figure 7, as well as the transmitting station identifiers (TOP and DDC), are such communications information and not part of the standard meteorological code. They do, however, identify the type and area covered by the message (UJ is a standard upper air report and US is self-explanatory). The B6500 would block all reports of the same type together.

5.2 Scanning and Recognition

Heimerdinger's procedure checks for tokens of the language by locating delimeters (in some cases the delimeters are tokens themselves). In the language of teletype reports, delimeters have no other meaning; the tokens are the five character data groups plus the identifying TT, VV, and QQ groups. The

```
NNNN

ZCZCWNELC449C 1446Z
UJUS KSTL 081410Z

TOP TT 58121 72456 99983 12222 36015 00124 ///// ///// 85473
05203 00531 70040 02738 30018 50565 16930 26531 40728 29345

25529 30927 457// 20041 20190 471// 24562 15378 533// 25555
10636 583// 25529

88231 571// 23072
77223 24076Φ


VV 5812/ 72456 00983 12222 15953 10415 22820 03600 33733

00100 44711 01505 55696 03149 66645 04345 77572 10313 88499
16930 99400 29348 11333 40147 22255 543// 33231 571// 44218

493// 55198 471// 66170 513// 77132 551// 88120 545// 99100
583// 31313 25046 549// 20558Φ


QQ 58120 72456 90023 36015 36026 36026 90467 00530 01023

36020 90979 33018 32018 91246 27518 26520 27027 92057 27535
25034 93056 20040 22064 24076 950// 24542Φ


DDC TT 58121 72451 99929 07220 01020 00169 ///// ///// 85509
00400 03031 70070 03905 29014 50566 19358 26522 40728 32561
26019 30924 479// 25021 20189 501// 24541 15376 527// 25048
10636 567// 25520

88278 515// 24528
77999Φ


VV 5812/ 72451 00929 07220 11913 04659 22887 02400 33850

00400 44824 03000 55742 00500 66701 03700 77680 05561 88669
04757 99584 10967 11529 15562 22480 21956 33400 32561 44353

39958 55278 515// 66178 497// 77100 567// 31313 35043 509//
24037Φ


QQ 58120 72451 90034 01020 01529 02038 90678 04023 04019

36010 909// 31012 91246 27516 26518 28018 92057 25526 26519
93057 25522 23544 9426/ 24037 25049 950// 25034Φ
```

Figure 7. Sample Teletype Data.

reports are stored in memory in increasing word addresses (i.e., across PEs), which is the easiest method of organization for an ILLIAC IV disk file [13]. Groups may be split across PE boundaries. Each five bit teletype signal is converted to a corresponding 8-bit code and one character is stored per byte. With the provision that delimeters are not tokens themselves and thus can be disregarded after the tokens are formed, a maximum of four tokens can be found in one 8-byte word.

The first routine in Appendix B indicates how the scanning and recognition begins. The routine is coded in the ILLIAC IV assembly language ASK [14]. At the label NEXTTTYLINE a row of 512 input characters is read from memory. The byte logical test operations are used to build a control string. A reset bit seven in each byte indicates a delimeter present and a reset bit six indicates a delimeter in the preceding byte. The symbol "Ø", which indicates the end of an independent part of meteorological code, is also searched for and marked in the control string.

At NEXTBYTE the process of examining the control string begins. The control bytes are successively loaded into Register D in order to set the mode pattern. This control of mode is used to control which PEs will be enabled to concatenate a new character onto the old partial group and which to store completed groups (tokens).

Any token not completed at the end of the word (a non-delimeter in the last byte) is kept for possible concatenation with an incomplete token in the next higher PE. After the comment SECTION JOINER, each PE routes to its predecessor the first token it found and the control byte. Only PEs with a non-delimeter last byte and a routed token are enabled for concatenation of the token. A routed token that is not concatenated (delimeter last byte) or that is too long (an error condition) is routed back and stored in the higher PE.

The question of efficiency for this routine is debatable. On one hand, all PEs participate in the frequent tests. However, on a serial machine transfer to another section of code would occur where disabling PEs occur on ILLIAC IV. Considering that on a long term average, PEs will be executing store operations for only three out of every eight bytes, efficiency of 20-30% might be reasonable. A time estimate for processing one row in this manner is 900 clock cycles or 45 microseconds*.

As a result of the processing so far described, five character groups have been recognized and stored left justified in PE memory. The storage locations for all PEs are a block of four words per PE.

The next step is to place all groups of one report into a single PE's memory. However, this process is more efficiently accomplished if most PEs have found a start of a message group. Thus the recognition of lines continues. Approximately fifty lines of 512 characters can be so processed.

Although not coded, the following algorithm will provide reasonably efficient code. Relocation of groups is accomplished as follows:

1) The CU causes each PE to test each group it has located for a "TT" group. The cummulative test results are transmitted to the CU and each PE with a TT group becomes a receiver for the remainder of that report.

2) The resulting pattern is used by the CU to control routes to the PEs that contain TT groups.

3) The index of the four word block (corresponding to one original row of teletype data) is routed to the next higher PE.

---

*All time estimates are based on a clock cycle of fifty nanoseconds.

4) The CU shifts the pattern of receiving PEs by one in order to locate the PEs that are to transmit groups; if PE 0 is one such PE (the route was from PE 63), only it is enabled to increment the fetch index by four (i.e., continue the message on the next row of memory).

5) The PEs with routed indices are enabled and route the four locations of the indicated block to the lower PE. If the receiving PE detects a TT group, it has completely received one report and is disabled from further stores.

6) The index is routed again to the next higher PE and the procedure loops to step 3 with a route of 2, then 3, etc.

7) The CU periodically samples the modes of all PEs and when all receivers are disabled, the procedure terminates.

It is estimated that this routing can be done with 50-70% efficiency. Based on an average routing time of four clock cycles and 100 groups per report, 70 to 100 reports could be rearranged in this manner in 100 microseconds.

5.3 Decoding

The rigid format of one particular type of report easily allows decoding in parallel. After the tests for a particular type group (perhaps for the third group of a TT report being 00 <digit> <digit> <digit>) PEs that pass the tests are disabled and failures recorded for use by error recovery routines. The successful PEs are then enabled and data portions of the group are converted to binary. At this stage, data are placed into 32-bit words with height and temperature in one word and the two wind components in another.

The determination of station location is based on directory comparison. After building in each PE an index table based on the last digit of the station numbers, the CU broadcasts the directory entries for each of the last digits. With 128 PEs, a typical set of 500 to 600 surface stations can be checked with only one or two comparisons per directory entry. It would be indeed unusual if ten stations of the same last digit were stored together in one PE.

Once station location is determined in i and j coordinates, the wind data for each reported level can be converted to the i and j components of geostrophic gradient. The sine of latitude, calculated to find the geostrophic components, is saved to be used in the calculations of non-geostrophic gradients during the analysis routines.

As the array of decoded data is built, the data actually present in a report is recorded for later use.

The basic recognition algorithm should be better than 50% efficient. Perhaps 25% of the total time would be spent waiting during memory fetches for which no successful match will occur. However, the directory lookup can be no more efficient than 1/128 because a station will only be found in one PE.

5.4 Hydrostatic Checking

The check of the report with the hydrostatic formula can be done in parallel on 128 stations at one time. However, as with all errors, at best invalid data can be flagged for later processing.

5.5 Error Recovery

The types of errors possible are many and the assumptions made concerning them will determine to what degree parallel operation can be effected. At best, only a few PEs would be involved for one particular type of error.

6. RELOCATION AND ANALYSIS OF DATA

6.1 Initial Assignment of Stations to PEs

As the controlling routine of the analysis procedure sequences to a new level, the data stored in the total report of each station must be extracted and passed to all PEs with points within 6.9 grid units of the station. The location of the station reports is totally a function of the original teletype message stream. However, broadcast of station location from the CU allows 128 simultaneous tests for the location. The data can then be broadcast with only PEs passing the location test enabled to store the data.

The most efficient way to transmit data to the CU is by the indexed block fetch (BINX) instruction. Preliminarily, each PE checks for the first station with data for the indicated level. Stations with both height and wind information are transferred first, then height only and wind only. The source of data for the BINX instruction is a skewed array formed by placing the location half word in array location 0 of the PE of origin, the sine of latitude in location 1 after a route of +1, the height and temperature in the location 2 after a route of +2, and the wind (height gradient) components in location 3 after a route of +3. The indices of the PEs are set so that one block fetch will transmit four complete sets of data (two 32-bit words in each location fetched).

As each station is received by a PE it is further checked for location within its area of responsibility or in the neighborhood area required for the consistency check; appropriate index lists are built.

6.2 Interpolated Values and Consistency Check

Before actual analysis can begin, the average field values for each station must be computed. The responsibility index list facilitates this

calculation. The sine of latitude is removed from its location and is replaced by the interpolated height and temperature values. The geostrophic wind components are also extracted and the two are saved in arrays ordered indentically to the responsibility list. The consistency check can then use the calculated averages as it compares for consistency of the observed data.

The interpolated field data is then transmitted to all other PEs using the station. This transfer requires routing first the location word (to identify the station) and then the interpolated data. The routing must be done for the rows of 16-point blocks up to +4 and -3 away and to the adjacent PEs of the same row. Routing between the inner half and outer half of the same PE is actually the execution of a SWAPA instruction, which interchanges the inner and outer words of the A register. A sequence of routes and interchanges will cause the station report in question to pass through many PEs (the differences in lengths of rows cause this situation). Broadcast of a mode pattern controls the store of the results of the distance test so that the proper mode test results can be restored when the interpolated data is routed and ready to be stored.

6.3 Interchange of Field Values

The first-guess field is initially loaded only into the PEs in which the calculation is to be done. Before each scan, field values must be transferred into all PEs requiring the points for consistency check and height gradient calculation. The method is essentially that of 6.2; however, it is programmatically known from and to which array locations the data transfer involves. Thus no tests are required; broadcast of appropriate mode patterns will enable only the desired stores after each route and interchange operation. This interchange is also simpler than that of 6.2 because routes to adjacent rows are the greatest required.

6.4 Analysis Calculations

Once the transfer of data to appropriate PEs is complete, the calculation of the analysis is straight-forward. Appendix B contains listings of two routines necessary to the actual analysis.

The routine STATIONSORT builds the array of pointers to stations to be used for analysis at each point. Note that the order of points is not that of their occurrence in the array, but rather, an order designed to improve computational efficiency. That is, the point within each PE's block of 16 for which the greatest number of stations are used in the correction computation is first and the one with the least number of stations is last. At the label AL1 in the procedure, one of the four bit pointers is extracted from the two word array POINTORDER (see Figure 8).

The first step accomplished with the data for a grid point is to calculate the distance from grid point to station (label AL2). Thereafter a sequence of tests compares this distance with each of the three radii of influence. For the lowest of the radii of influence, RLOW, and data with both height and wind reports, stations are stored directly into the pointer array STAPERPNT. For the other combinations of radii and reported data, temporary arrays are built. After all stations with height reports are checked, wind only reports are processed.

The array STAPERPNT packs four 8-bit pointers to each word. Since the pointers are to four word blocks in a PE's memory, they must be left shifted before use. Also, in order to use an all zero pointer as a delimeter between types of data, the pointers are initially set one block (four rows of PE memory) high.

The constant checking of the same stations against points, obviously leads to some computational inefficiency. However, the routine that assigns

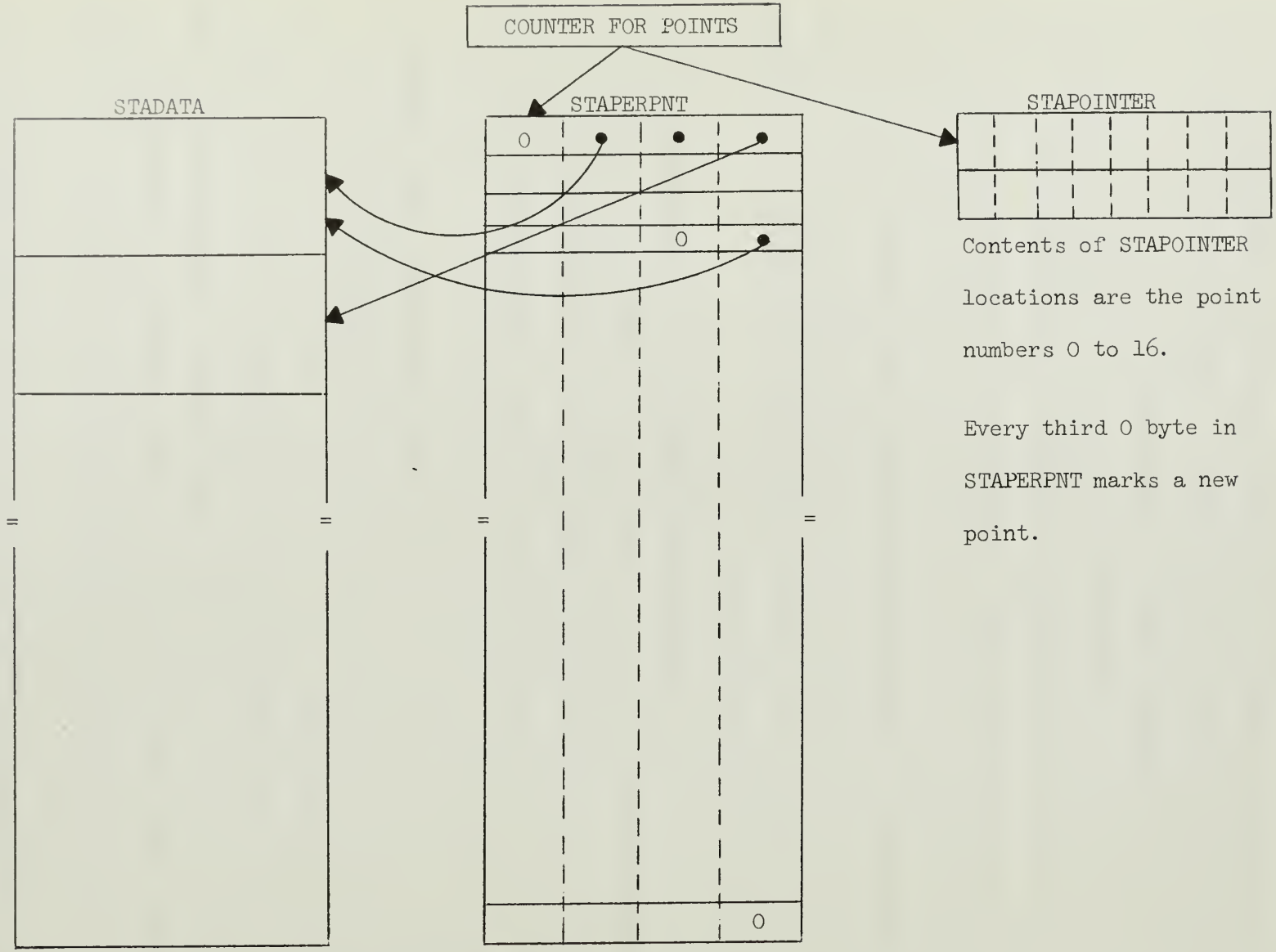


Figure 8. Data Organization During Analysis

stations to grid points has a beneficial effect here--all points do not have to be checked against all stations. An estimate of 30-50% efficiency is reasonable. With the possibility some PEs have 200 stations, the time for this procedure approaches one fourth of a second.

The routine SCANFIELD, however, is highly parallel. The procedure UNPACK takes pointers to STAPERPNT and locates the station data into temporary storage. The computation that follows is a straight-forward application of the analysis formulas. The building of STAPERPNT was such that the computations of the three types of corrections are done sequentially. The final computed value replaces the previous first-guess value.

The efficiency of computation begins at 100% as every block will contain some data. However, it will quickly drop off. The distribution of data will probably force efficiency under 50%. The time of one scan over the field is estimated to be 75 milliseconds.

6.5 Analyst Interaction

During the execution of the system presently employed at NMC, an analyst monitors the machine analysis. In particular, he has the option of forcing into the analysis bogus or doubtful data. Such procedures would be wasteful of ILLIAC IV processing power. Considering the speed with which one scan can be made, it is suggested that more than one complete set of analyses can be computed, with the analyst inserting bogus data between runs in order to control the results as desired.

7. CONCLUSIONS

The weather analysis problem is primarily one of much data, assymmetrically distributed, and a few calculations. Algorithms for storage of unevenly distributed data can be developed for ILLIAC IV. The handling of input data and calculation of the analysis can be done with a great deal of parallelism. The most difficult, and time-consuming, steps are those requiring broadcast of data from the CU. More detailed knowledge of the input data may lead to more efficient means of directory look-up and of assignment of station reports to grid points.

Considering that several time-consuming routines have not been coded, an estimate for total analysis time for one level is five to ten seconds.

LIST OF REFERENCES

[1] Thompson, P. D. Numerical Weather Analysis and Prediction, McMillan Co., New York, 1961.

[2] Hess, S. L. Introduction to Theoretical Meteorology, Holt, Rinehart, and Winston, New York, 1959.

[3] Beyers, R. B. General Meteorology, third edition, McGraw-Hill, New York, 1959.

[4] Gilchrist, B., and Cressman, G. P. "An Experiment in Objective Analysis", Tellus, vol. 6, no. 4, Nov. 1954, pp. 309-318.

[5] Bergthorsson, P., and Doos, B. "Numerical Weather Map Analysis", Tellus, vol. 7, no. 3, Aug. 1955, pp. 329-340.

[6] Cressman, G. P. "An Operational Objective Analysis System", Monthly Weather Review, vol. 87, no. 10, Oct. 1959, pp. 367-374.

[7] McDonell, J. E. "Notes on NMC Objective Analysis Programs", Data Automation Division, National Meteorological Center, Oct. 1968, 9 pages plus attachments (unpublished paper).

[8] Bedient, H., Collins, W. G., and Dent, G. "An Operational Tropical Analysis System", Monthly Weather Review, vol. 95, no. 12, Dec. 1967, pp. 942-949.

[9] Burroughs Corporation, ILLIAC IV Systems Characteristics and Programming Manual, with Change 2, Jan. 20, 1969.

[10] Westlund, G. "A Timing Simulator for ILLIAC IV", ILLIAC IV Document No. 202, Department of Computer Science, University of Illinois, Sept. 6, 1969.

[11] Air Weather Service, Meteorological Codes, Manual 105-24, July 15, 1968.

[12] Heimerdinger, W. L. "String Processing on a Parallel Computer", M.S. Thesis, University of Illinois, Feb. 1969 (also, Department of Computer Science Report No. 299, Jan. 1969).

[13] Alsberg, P. A., et al. "A Description of ILLIAC IV Operating System", ILLIAC IV Document No. 212, Department of Computer Science, University of Illinois, Nov. 1, 1969.

[14] Grothe, D., and Luskin, C. Reference Manual for ILLIAC IV Assembler (ASK), Burroughs Corporation, Mar. 17, 1969.

APPENDIX A

SYNTAX FOR UPPER AIR REPORTS FOR LAND STATIONS

1. Report Structure

<upper air report for land station> ::= <upper level report for land station>
<line delimiter>
<upper wind report for land station>;

<upper level report for land station> ::= <standard level report>
<line delimeter>
<significant level report>.

2. Standard Level Report Structure

<standard level report> ::= <identification section 1> <delimeter>
<surface data> <delimeter> <standard level data>
<delimeter> <tropopause report> <delimeter>
<maximum wind report> Ø ;

<identification section 1> ::= TT' <delimeter> <date-time indicator>
<wind levels indicator> <delimeter>
<station designator>

<surface data> ::= 99<pressure> <delimeter> <temperature group> <delimeter>
<wind group>;

<standard level data> ::= <level and height group> <delimeter>
<temperature group> <delimeter> <wind group>;

<level and height group> ::= <level indicator> <height>;

<level indicator> ::= 00 | 85 | 70 | 50 | 40 | 30 | 20 | 15 | 10 ;

<height> ::= <digit> <digit> <digit>;

<tropopause report> ::= <tropopause data> | <tropopause report> <delimeter>
<tropopause data> | 88999;

<tropopause data> ::= <tropopause group> <delimeter> <temperature group>
<delimeter> <wind group>;

<tropopause group> ::= 88<pressure>;

<maximum wind report> ::= <top of sounding report> | <internal maximum report> |
77999 ;

<top of sounding report> ::= <top of sounding group> <delimeter> <wind group>;

<top of sounding group> ::= 66<pressure>;

<internal maximum report> ::= <internal maximum data> | <internal maximum report>
<delimeter> <internal maximum data>;

<internal maximum data> ::= <internal maximum group> <delimeter> <wind group>;

<internal maximum group> ::= 77<pressure>.

3. Significant Level Report Structure

<significant level report> ::= <identification section 2> <delimeter>
<significant temperature level report> <delimeter>
<250 milibar report> <delimeter>
<regional code section> 0 ;

<identification section 2> ::= VV<delimeter> <date time indicator>/<delimeter>
<station designator>;

<significant temperature levels report> ::= <surface temperature><delimeter>
<significant temperature levels data>;

<surface temperature> ::=

<significant temperature levels data> ::= <significant temperature level group>
<delimeter> <temperature group> |
<significant temperature levels data>

<delimeter>

<significant temperature level group>

<delimeter> <temperature group>;

<significant temperature level group> ::= <significant level indicator>

<pressure>;

<significant level indicator> ::= 11 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99 ;

<250 milibar report> ::= 31313<delimeter> 25<height> <delimeter>

<temperature group> <delimeter> <wind group>;

<regional code section> ::= 51515<delimeter> <regional code groups>;

<regional code groups> ::= <101 group> | <super remarks>;

<101 group> ::= 101<101 code>;

<101 code> ::= <digit> <digit>;

<super remarks> ::= SUPER<delimeter> <super layers>;

<super layers> ::= <digit> <digit> - <digit> <digit> | <super layers> <digit>

<digit> - <digit> <digit>.

4. Upper Wind Report Structure

<upper wind report for land station> ::= <regional standard level report> 0 ;

<regional standard level report> ::= <identification section 3> <delimeter>

<regional standard level data>;

<identification section 3> ::= QQ<delimeter> <date time indicator>

<equipment indicator> <delimeter>

<station designator>;

<equipment indicator> ::= <digit>;

<regional standard level data> ::= <level groups> |

<regional standard level data> <delimeter>

<level groups>;

<level groups> ::= <single level groups> | <double level groups> |
<triple level groups>;

<single level groups> ::= 9<tens digit> <ones digit> // <delimeter> <wind group>;

<double level groups> ::= 9<tens digit> <ones digit> <ones digit> / <delimeter>
<wind group> <delimeter> <wind group>;

<triple level groups> ::= 9<tens digit> <ones digit> <ones digit> <ones digit>
<delimeter> <wind group> <delimeter> <wind group>
<delimeter> <wind group>;

<tens digit> ::= <digit>;

<ones digit> ::= <digit>.

5. General Terminal Symbols

<delimeter> ::= <line delimeter> | <blank>;

<line delimeter> ::= {combination of the carriage return and line feed teletype
control symbols that cause typing to begin on a new line};

<blank> ::= {teletype impulse that spaces the typing equipment without printing
a character}.

6. General Non-Terminal Symbols

<date-time indicator> ::= <date and units code> <time>;

<date and units code> ::= <digit> <digit>

<time> ::= 00 | 12 ;

<wind levels indicator> ::= <digit> ;

<station designator> ::= <block number> <station number> ;

<block number> ::= <digit> <digit>;

<station number> ::= <digit> <digit> <digit>;

<pressure> ::= <digit> <digit> <digit>;

<temperature group> ::= <temperature in degrees> <tenths and sign indicator>
<dew point depression> | /////;

<temperature in degrees> ::= <digit> <digit> <digit>;

<tenths and sign indicator> ::= <digit>;

<dew point depression> ::= <digit> <digit>;

<wind group> ::= <wind direction in tens> <five degrees and hundreds of speed>
<speed> | /////;

<wind direction in tens> ::= <digit> <digit>;

<five degrees and hundreds of speed> ::= <digit>;

<speed> ::= <digit>.

7. Metalinguistic Symbols

Diamond brackets group the descriptions of non-terminal symbols. The ::= is the replacement operator and | indicates "or". Semicolons indicate the end of a metalinguistic statement and periods, of a section.

APPENDIX B

```
BEGIN                                                                   0000010
* DEFINES FOR CU REGISTERS                                              0000020
                DEFINE CARBDC=$C0**;                                    0000030
                DEFINE CARALL=$C1**;                                    0000040
MAXBLKSATTY;    EQU  100;          * ONLY AN ESTIMATE                   0000050
STEPUNTIL56;    EQU   =8,56,0;                                          0000060
RSAR62;         EQU   0FFFFFFFFFFFFFFFD;16;  * MASK FOR RESET BY AND   0000070
.BDC;           EQU   CARBDC;       * FOR BROADCASTING DATA             0000080
.ALL;           EQU   CARALL;      * ALL BITS SET FOR MODE BROADCAST    0000090
.BYTESTEP;      EQU   $C2;         * FOR LOOP CONTROL IN SETION BLDR    0000100
.SECTNO;        EQU   $C3;         * # OF GROUPS OF 4BLKS FOR SECTNS    0000110
.TTYSTEP;       EQU   $C3;          * FOR SEQUENTIAL CONTROL            0000120
.HEADER;        EQU   $D0;          * FOR SEQUENCE HEADER PATTERN       0000130
.CRPAT;         EQU   $D1;          * FOR CARRAIAGE RETURN CHARACTER    0000140
.LFPAT;         EQU   $D2;          * FOR LINE FEED CHARACTER           0000150
.BLKPAT;        EQU   $D3;          * FOR BLANK CHARACTER               0000160
.ENDPAT;        EQU   $D4;         * FOR END-OF-MESSAGE SIGNAL          0000170
.BYTESTEPSTORE; EQU   $D5;         * LOCAL STORAGE FOR STEPBUNTIL56     0000180
.COMMIT;        EQU   $D6;         * RESET BIT62 IN COUNTER             0000190
.ENABLEBITS;    EQU   $D7;         * SET BITS FOR E,E1 IN CONTROL WD    0000200
.TTYBLKSTORE;   EQU   $D63;                                             0000210
* INITIALIZE CU AND PE REGISTERS                                        0000220
        SLIT(0)   CHARPATS;                                             0000230
        BIN(0)    $D0;              * CHARACTER PATTERNS TO ADR          0000240
        CLC       (.ALL);                                               0000250
        COMPC     (.ALL);                                               0000260
        LDFE1     CARALL;          * ALL ON                             0000270
        CLRA;                                                           0000280
        LDS       $A;               * CLEAR S                         0000290
        LDX       $S;              * CLEAR RGX                          0000300
        LIT (.TTYSTEP) =64,64*MAXBLKSATTY,0;   * FOR SEQUENTIAL CON-    0000310
                                   *   TROL OF PROCESSING OF INPUT      0000320
NEXTTTYLINE;                                                            0000330
        LDA       TTYREPORTS(.TTYSTEP); * 512 INPUT CHARACTERS          0000340
        STL(.TTYSTEP).TTYBLKSTORE;                                      0000350
        LDR       $A;               * SAVE COPY OF STRING              0000360
        LIT(.BDC) =1,SECTIONS+3*64,SECTIONS;                            0000370
        DISPLAY   .BDC,31;                                              0000380
        LDL(0)    .CRPAT;                                               0000390
        NER       CARBDC;           * FIND CR S                       0000400
        LDS       $A;              * BIT 7 OF EACH BYTE MARKS NOT      0000410
                                   * DELIMETER                          0000420
        LDA       $R;                                                   0000430
        LDL(0)    .LFPAT;                                               0000440
        NER       CARBDC;           * FIND LF S                       0000450
        AND       $S;              * TURN OFF NOT-DELIMETER MARKERS     0000460
                                   * FOR ANY LF S                       0000470
        LDS       $A;                                                   0000480
        LDA       $R;                                                   0000490
        LDL(0)    .BLKPAT;                                              0000500
        NER       CARBDC;          * FIND BLANKS                       0000510
        AND       $S;              * TURN OFF NOT-DELIMETER MARKERS     0000520
                                   * FOR ANY BLAKS                      0000530
        LDA       $R;                                                   0000540
```

```
        LDS     $A;                 % SAVE FOR CURRENT BYTE MARKER   0000550
        SHAR    6;                  % PRECEEDING BYTE NO-DELIMETER   0000560
        OR      $S;                 %   AND INTO CONTROL STRING     0000570
        LDS     $A;                                                 0000580
        LDA     $R;                 % SOURCE STRING                  0000590
        LDL(0)  .FOMPAT;                                            0000600
        NEB     CARBDC;               % FIND EOM SIGNALS             0000610
        RTAR    5;                                                  0000620
        OR      $S;                 % MARKER INTO MARKER WORD        0000630
        LDL(.BDC) .ENABLEBITS;      % SO THAT ALL PES WILL BE INITIA-0000640
        OR      CARBDC;               %   LY ENABLED AFTER EACH LDD  0000650
        STA     MARKER;             % SAVE CONTROL STRING            0000660
% CONTROL STRING NOW FORMED IN EACH PE; NOW  SET UP SECTION BUILDER   0000670
        LDFE1   CARALL;             % ALL PES ON                     0000680
        LDL(.BYTESTEP),BYTESTEPSTORE;                               0000690
        SKIP    ,2;                 % SKIP RE-INITIALIZATION         0000700
NEXTBYTE;LDA    MARKER;                                             0000710
        RTAL    0(.BYTESTEP);       % FIND PROPER CONTROL BYTE       0000720
        LDR     $A;                                                 0000730
        LDD     $B;                 % CONTROL BYTE INTO RGD          0000740
        LDA     $R;                 % SOURE STRING FOR EXTRACTION OF 0000750
                                    %   NEXT BYTE                   0000760
        SHAR    56;                 % EXTRACT NEXT  CHARACTER        0000770
        RTAR    #8;                 %   AND PLACE IN POSITION       0000780
        OR      $B;                 %   AND INTO PARTIAL SECTION    0000790
        SWAP;                       % OLD PARTIAL SECTION TO RGA     0000800
        SETE    J AND E;            % ENABLE ONLY THOSE PES ADDING TO0000810
        SETE1   J AND E;            %   PARTIAL SECTION             0000820
        LDA     $B;                 % RGA CHANGED IN PES ADDING BYTE 0000830
        LDR     $S;                 % SWITCH RGX & RGS               0000840
        LDS     $X;                 %   STORE INDEX  TO RGS         0000850
        LDX     $B;                 %   COUNTER TO RGX              0000860
        XI      =8;                 % INCREMENT COUNTER              0000870
        LDR     $S;                 % SWITCH RGS & RGX               0000880
        LDS     $X;                 %   COUNTER TO RGX              0000890
        LDX     $B;                 %   STORE INDEX TO RGX          0000900
        LDFE1   CARALL;             % ALL BACK ON                    0000910
        SETE    -J AND E;           % ENABLE ONLY THOSE WITH A DELI- 0000920
        SETE1   -J AND E;           %   METER                       0000930
        SETE    H AND E;            % AND WITHOUT A PRECEEDING DELI- 0000940
        SETE1   H AND E;            %   METER                       0000950
        SETI    H AND E;            % I=1 FOR PES WITH SECTION TO    0000960
                                    %   STORE                       0000970
        OR      $S;                 % COUNTER INTO LAST BYTE OF SEC- 0000980
                                    %   TION                        0000990
        SETE    G AND E;            % ENABLE ONLY THOSE PES WITH EOM 0001000
        SETE1   G AND E;            %   CHARACTER IN PRCEEDING BYTE 0001010
        OR      =2;                 %   INTO COUNTER BYTE           0001020
        LDFE1   CARALL;             % RE-ENABLE ALL PES WITH PARTIAL 0001030
        SETE    I  AND E;           %   SECTIONS TO STORE           0001040
        SETE1   I  AND E;                                           0001050
        STA     *SECTIONS(.SECTNO); % STORE COMPETED SECTION        0001060
        XI      =1;                 % INCREMENT REGX                 0001070
        CLRA;                       % NEW NULL PARTIAL SETION        0001080
        LDS     $A;                 %   & COUNTER                   0001090
        LDFE1   CARALL;             % ENABLE ALL                     0001100
        LDR     $A;                 % PARTIAL SECTION                0001110
        LDA     $P;                 % SOURCE STRING                  0001120
        SHAL    8;                  % REMOVE BYTE JUST PROCESSED     0001130
        LDR     $A;                 % REMAINING SOURCE BYTES IN RGR  0001140
        TXLTM(.BYTESTEP) ,NEXTBYTE;                                 0001150
```

```
% SECTION JOINER                                                    0001160
        RTAL    #0;                     % RIGHT JUSTIFY PARTIAL SECTION  0001170
        LDB     MARKER;                                                0001180
        LDD     $B;                     % FIRST CONTROL BYTE             0001190
        LDR     $A;                     % PARTIAL SECTION TO RGR         0001200
        LDA     $S;                     % COUNTER MOVED TO RGA           0001210
        SETE    -G AND E;               % ENABLE ONLY THOSE PFS WITH EOM 0001220
        SETE1   -G AND E;               %   CHARACTER IN LAST BYTE       0001230
        OR      #2;                     % MARKER INTO COUNTER BYTE       0001240
        LDEE1   CARALL;                 % ALL ON                         0001250
        STX     T1;                     % TEMPORARY STORAGE              0001260
        LDX     $S;                     % COUNTER TO RGX                 0001270
        RTL     1;                      % PARTIAL SECTION ROUTED         0001280
        LDS     $R;                     %   INTO RGS OF NEXT HIGHER PE   0001290
        RTL     $A,1;                   % PARTIAL SECTION COUNTER ROUTED 0001300
        LDA     $R;                     % ROUTED COUNTER TO RGA          0001310
% FIRST CASE: DELIMETER IN 1ST BYTE IMPLIES NO CHANGE TO ROUTED SECTION  0001320
        SETE    J AND E;                % CASE 1 PFS DISABLED           0001330
        SETE1   J AND E;                                               0001340
% SECOND CASE: PE ALREADY HAS 8 NON-DELIMETER CHARACTERS                0001350
        IXE     64;                     % I=1 FOR CASE 2 PFS            0001360
        SETG    -I AND E;               % G=1 FOR CASE 3 PFS            0001370
        SETE    I AND E;                % ENABLE ONLY CASE 2 PFS        0001380
        SETE1   I AND E;                                               0001390
        OR      #1;                     % CONTINUATION MARK IN BIT63    0001400
        LDEE1   CARALL;                 % ALL ON                        0001410
% THIRD CASE: COMPOUND PARTIAL SECTIONS ON BASIS OF COUNTERS             0001420
        SETE    G AND E;                % ENABLE ONLY CASE 3 PFS        0001430
        SETE1   G AND E;                                               0001440
        LDL(.BDC) .EOMOUT;                                             0001450
        AND     CARBDC;                 % EOM CHARACTER IS NOT LAST BYTE 0001460
        IDR     $A;                     % ROUTED COUNTER                0001470
        LDA     SECTIONS(.SECTNO);      % 1ST SECTION TO RGA            0001480
        SHABR   8;                      % EXTRACT COUNTER               0001490
        SWAP;                                                          0001500
        SHAR    56;                     % INTO LOW-ORDER BYTE           0001510
        SWAP;                           % PARTIAL SECTIO SAVED IN RGA   0001520
        LDX     #56;                                                   0001530
        XD      $A;                     % BITS THAT CAN BE ADDED FROM   0001540
        LDB     $R;                                                    0001550
        LDB     $R;                     % ROUTED SECTION TO RGR         0001560
        SHAL    A;                      % RE-ALIGN PARTIAL SECTION      0001570
        SWAP;                                                          0001580
        SHABR   #0;                     % CONCATENATE TWO SECTIONS      0001590
        STR     SECTIONS+3;                                            0001600
        TXLTM(.TTYSTEP),NEXTTTYLINE;                                   0001610
        HALT;                                                          0001620
             FILL 8;                                                   0001630
CHARPATS:    DATA "TT000002","<<<<<<<<","########","        ","@@@@@@@@"0001640
             "@",STEPUNTIL56,RSAB62,0C0C0C0C0C0C0C0C0C0;16;            0001650
MARKER:      BLK  1;                                                   0001660
T1:          BLK  1;                 % TEMPORARY STORE               0001670
TTYREPORTS:  BLK  MAXBLKS4TTY;                                         0001680
SECTIONS:    BLK  4*MAXBLKS4TTY;                                       0001690
END.                                                                   0001700
```

```
BEGIN                                                                   0000010
% STATIONSORT COMPARES THE DISTANCE FROM EACH POINT TO EACH STATION     0000020
% IN THAT BLOCK. IT FORMS THE INDEX LIST STAPERPNT, WHICH IS USED IN    0000030
% THE ANALYSIS ROUTINE TO DETERMINE WHICH STATIONS ARE TO BE USED TO    0000040
% ADJUST THE FIRST GUESS FIELD. THE THREE TYPES OF REPORTS ARE GROUPED  0000050
% TOGETHER IN THE LIST FOR EACH STATION. THE TYPES, AS WELL AS THE      0000060
% POINTS THEMSELVES,ARE DELIMETED BY AN ALL ZERO POINTER. THUS TO COM-  0000070
% PUTE THE ROW ADDRESS OF A STATION, SUBTRACT 1 FROM THE ADDRESS OF THE 0000080
% ARRAY.                                                                0000090
% DEFINE STATEMENTS FOR CU REGISTERS                                    0000100
                    DEFINE CARBDC=$C0##;                                0000110
                    DEFINE CARALL=$C1##;                                0000120
% INITIALIZATION                                                        0000130
          LDEF1     CARALL;             % ENABLE ALL PES               0000140
          CLRA;                                                         0000150
          LIT(.CNTR)=64,127*64,0;                                       0000160
          STA       NEXTSTORE;          % ZERO OUT STORE POINTER       0000170
          STA       NEXTSHIFT;                                          0000180
CLRSTAPERPNT: STA   STAPERPNT(.CNTR);   % UNUSED PORTIONS MUST BE 0 WHEN 0000190
                                        %   FIRST POINTERS ARE STORED   0000200
          TXLTM(.CNTR),CLRSTAPERPNT;                                    0000210
          LIT(.CNTR)=64,15*64,0;        % 0 STEP 1 UNTIL 16 IN ROW     0000220
                                        %   ADDRESSES                   0000230
          CSB(.FLAG).FIRSTHALF;         % MARK THAT USING 1ST HALF OF  0000240
                                        %   POINTORDER                  0000250
          LDL(.BDC) .STEPM4UNTIL0;      % INDEX TO UNPACK POINTORDER   0000260
% INITIALIZATION COMPLETE                                               0000270
          LDA       POINTORDER;                                       0000280
AL1:      SHAP      0(.BDC);            % NEXT POINTER TO LOW ORDER BITS 0000290
          ALIT(.BDC)=-4;                                                0000300
          STL(.BDC) .STORESTEPM4;       % SAVE SHIFT COUNTER           0000310
          LDL(.BDC).MASKOUTOTO28;       % FOR ANDING OUT HIGH ORDER BITS 0000320
          AND       CARBDC;             % ZERO BITS 0 TO 28            0000330
          LDL(.BDC) .PAIR8;             % FOR TEST OF ROW              0000340
          IML       CARBDC;             % TEST FOR ROW OF POINT        0000350
          SETF      -I AND E;           % DISABLE WHEN 1ST ROW         0000360
          SETF1     -G AND E1;                                          0000370
          SBM       CARBDC;             % COLUMN WITHIN BLOCK FOR 2ND ROW0000380
          LDEF1     CARALL;                                           0000390
          SHAL      4;                  % SCALE TO 1/16               0000400
          ADM       ISTART;             % I-COORDINATE OF POINT        0000410
          STA       ICURRENT;           % SAVE I-COORDINATE            0000420
          LDA       JSTART;             % FOR COMPUTING J-COORDINATE   0000430
          SETF      -I AND E;           % DISABLE IF FIRST ROW         0000440
          SETF1     -G AND E;                                         0000450
          CSHR(.BDC)1;                  % 1 SCALED TO 1/16             0000460
          ADM       CARBDC;             % J-COORDINATE SCALED TO 1/16  0000470
          LDEF1     CARALL;             % ENABLE ALL PES               0000480
          STA       JCURRENT;           % SAVE J-COORDINATE            0000490
% INITIALIZE COUNTERS FOR NEXT POINT                                    0000500
          LIT(.BDC) =64,8*64,0;                                         0000510
CLRCNTRS: STA       LOWBOTH(.BDC);                                      0000520
          TXLTM(.BDC),CLRCNTRS;                                         0000530
% FORM LOOP CONTROL WORD                                               0000540
          LIT(.CNTR)=1,-1,0;                                            0000550
          CSHR(.CNTR)24;                                                0000560
          CADD(.CNTR).MAXBOTH;                                          0000570
          CSHL(.CNTR)30;                                                0000580
```

```
% BEGIN LOOP TO CHECK STATIONS                                          0000590
AL2;    LDA     STADATA(.CNTR);                                        0000600
%COMPUTE (ISTATION-IPOINT)*2+(JSTATION-JPOINT)*2=DISTANCE SQUARED       0000610
        LDS     $A;                                                    0000620
        SHAR    16;             % I COORDINATE OF STATION SCALE        0000630
                                %   TO 1/16;                           0000640
        SBM     ICURRENT;       % ISTATION-IPOINT;SCALE 1/16           0000650
        MLM     $A;             % (ISTATION-IPOINT)*2 SCALE 1/256      0000660
        LDR     JCURRENT;       % RETRIEVE CURRENT IPOINT              0000670
        LDR     $A;             % (ISTATION-IPOINT)*2 SAVED            0000680
        LDA     $S;             % TO FIND JSTATION                     0000690
        LDS     $B;                                                    0000700
        LDL(.BDC) .MASKOUTOTO15;                                       0000710
        AND     CARBDC;         % JSTATION;SCALE 1/16                 0000720
        SBM     $B;             % JSTATION-JPOINT;SCALE 1/16           0000730
        MLM     $A;             % (JSTATION-JPOINT)*2;SCALE 1/256      0000740
        ADM     $S;             % DISTANCE SQUARED;SCALE 1/256         0000750
        ALIT(.CNTR)=64*4;       % INCREMENT ROW ADRESS BY 4            0000760
        LDL(.CNTR).CNTRSTORE;                                          0000770
        DUPD(.BDC).CNTRSTORE;   % FOR CALCULATION OF POINTER           0000780
        LDA     CARBDC;                                                0000790
        SHAR    6;              % CONVERT TO ROW INDEX                 0000800
        LDR     $A;             % POINTER TO RGR                       0000810
        IMG     STARTHT;        % TEST FOR END OF TYPE BOTH            0000820
        LDR     $A;                                                    0000830
        SETE    -I OR E;        % ENABLE THOSE WITH DATA               0000840
        SETE1   -G AND E1;                                             0000850
% DETERMINE IF STATION IS WITHIN RLOW OF POINT                         0000860
        LDL(.BDC) .RLOWSQ;      % LOWWEST RADIUS OF INFLUENCE          0000870
        JAL     CARBDC;         %   COMPARED WITH DISTANCE            0000880
        SETE    J AND E;        % DISABLE IF STATION OUTSIDE OF        0000890
        SETE1   H AND E;        %   RLOW                               0000900
        SLIT(.FLAG)=FORMPOINTER;                                       0000910
        EXCHL(.FLAG)$ICR;       % TRANSFER                            0000920
        LDS     $A;             % SAVE DISTANCE SQUARED                0000930
% UPDATE NUMBER OF REPORTS IN LOW RADIUS                               0000940
        LDL(.BDC) .PAIR1;                                              0000950
        LDA     CARBDC;                                                0000960
        ADM     LOWBOTH;                                               0000970
        STA     LOWBOTH;                                               0000980
        SETE    -I AND E;       % DISABLE PES WITH NO MORE STA-       0000990
        SETE1   -G AND E1;      %   TIONS OF TYPE BOTH                 0001000
% DETERMINE IF STATION IS WITHIN RMID OF POINT                         0001010
        LDL(.BDC) .RMIDSQ;      % RMID*2;SCALE 1/256                   0001020
        JML     CARBDC;         % IS STATION WITHIN MIDDLE RADIUS      0001030
        SETE    J AND E;        % DISABLE THOSE OUTSIDE OF RMID        0001040
        SETE1   H AND E1;                                              0001050
% IF STATION IS WITHIN RMID, PLACE POINTER IN TEMPORARY ARRAY          0001060
        LDA     MIDBOTH;        % FIND INDICES FOR TEMPORARY AR-       0001070
                                %   RAY                                0001080
        LDL(.CNTR),ADFMID;      % ADDRESS OF TEMPORARY ARRAY           0001090
        SLIT(.FLAG)=TEMPPOINTER;                                       0001100
        EXCHL(.FLAG)$ICR;       % JUMP TO TEMPOINTER                   0001110
        STA     MIDBOTH;        % STORE UPDATED POINTER                0001120
% DETERMINE IF STATION IS WITHIN RMAX OF POINT                         0001130
        LDL(.BDC).RUPPSQ;                                              0001140
        JML     CARBDC;                                                0001150
        SETE    J AND E;        % DISABLE THOSE OUTSIDE OF RMAX        0001160
        SETE1   H AND E1;                                              0001170
% IF STATION IS WITHIN RMAX, PLACE POINTER IN TEMPORARY ARRAY          0001180
        LDA     UPROTH;         % INDICES TO TEMPORARY ARRAY           0001190
```

```
         LDL(.CNTR),ADFUP;          % ADDRESS OF TEMPORARY ARRAY      0001200
         SLIT(.FLAG)=TEMPPOINTER;                                    0001210
         EXCHL(.FLAG)$ICR;          % JUMP TO TEMPPOINTER              0001220
         STA        UPBOTH;                                          0001230
         LDL(.CNTR),CNTRSTORE;                                       0001240
         TXLT(.CNTR),AL3;           % CHECK NEXT STATION               0001250
% DETERMINE RADIUS OF INFLUENCE BASED ON REPORTS OF TYPE BOTH          0001260
         LDA        NRLOW;                                           0001270
         LDL(.BDC) .PAIRNLM1;       % ARE THERE NL OR MORE REPORTS     0001280
         IMG        CARBDC;         %   WITHIN RLOW                  0001290
         SETF       I AND F;        % ENABLE ONLY THOSE PES WITH NL    0001300
         SETF1      G AND F1;       %   OR MORE                      0001310
         LDL(.BDC) .RLOWSQ;                                          0001320
         LDA        CARBDC;         % STORE RADIUS TO BE USED IN       0001330
         STA        RADIUSSQ;       %   CHECK OF HEIGHT ONLY REPORTS 0001340
         STL(.FLAG),LINK;                                            0001350
         SETF       -F OR -F;       % DISABLE RLOW PES                 0001360
         SETF1      -F1 OR -F1;     % AND ENABLE ALL OTHERS            0001370
         ADM        MIDBOTH;                                         0001380
         LDL(.BDC) .PAIRNMM1;       % ARE THERE NM OR MORE REPORTS     0001390
         JMG        CARBDC;         %   WITHIN RMID                  0001400
         SETF       J OR F;         % ENABLE ONLY THOSE PES WITH       0001410
         SETF1      H OR F1;        %   NM OR MORE                   0001420
         LDL(.BDC) .RMIDSQ;         % STORE RADIUS                     0001430
         LDA        CARBDC;                                          0001440
         STA        RADIUSSQ;                                        0001450
         SETF       -I AND -F;      % ENABLE PES THAT HAVE NOT STORED0001460
         SETF1      -G AND -F1;     %   A RADIUS                     0001470
         LDL(.BDC) .RMAXSQ;         % ALL REMAINING POINTS USE RMAX    0001480
         LDA        CARBDC;         %   FOR RADIUS                   0001490
         STA        RADIUSSQ;                                        0001500
% FIND INDEX TO HEIGHT ONLY STATIONS                                  0001510
         LDFF1      CARALL;         % ENABLE ALL PES                   0001520
         LDA       STARTHT;                                          0001530
         LDR        $A;             % OUTER INDEX TO RGX               0001540
         LDX        $R;                                              0001550
         SWAPA;                                                      0001560
         LDS       $A;              % INNER INDEX TO RGS               0001570
         SWAPA;                     % FOR INCREMENT                   0001580
         LDL(.BDC) .PAIR4;          % INCREMENT INDEX BY 4             0001590
         ADM        CARBDC;           %   FOR USE AS POINTER         0001600
         LDR        $A;               % POINTER IN PLACE             0001610
         SKIP       ,AL4+1;                                          0001620
%                                                                    0001630
% START OF LOOP FOR CHECK OF HEIGHT ONLY REPORTS                      0001640
AL4;      LDA        $R;              % INDEX IS IN RGR              0001650
% CHECK FOR END OF HEIGHT ONLY STATIONS                               0001660
         IMG        STARTWND;                                        0001670
         SETF       -I OR F;        % ENABLE THOSE WITH DATA           0001680
         SETF1      -G AND F1;                                       0001690
% DETERMINE IF STATION IS WITHIN RLOW OF POINT                        0001700
         LDL(.BDC) .RLOWSQ;         % LOWWEST RADIUS OF INFLUENCE      0001710
         JAL        CARBDC;         %   COMPARED WITH DISTANCE       0001720
         SETF       J AND F;        % DISABLE IF STATION OUTSIDE OF    0001730
         SETF1      H AND F;        %   RLOW                         0001740
% IF STATION IS WITHIN RLOW, PLACE POINTER IN TEMPORARY ARRAY         0001750
         LDL(.CNTR),ADFLOW;         % ADDRESS OF TEMPORARY ARRAY       0001760
         SLIT(.FLAG)=TEMPPOINTER;                                    0001770
         EXCHL(.FLAG)$ICR;          % JUMP TO TEMPOINTER               0001780
         STA        LOWPOINTER;     % STORE UPDATED POINTER            0001790
         LDA        $S;             % TEMPPOINTER PLACED DISTANCE      0001800
```

```
                                    % SQUARED IN RGS                    0001810
% DETERMINE IF STATION IS WITHIN RMID OF POINT                           0001820
        LDL(.BDC) .RMIDSQ}          % RMID*2;SCALE 1/256                  0001830
        JML       CARBDC}           % IS STATION WITHIN MIDDLE RADIUS     0001840
        SETF      J AND E}          % DISABLE THOSE OUTSIDE OR RMID       0001850
        SETF1     H AND F1}                                               0001860
% IF STATION IS WITHIN RMID, PLACE POINTER IN TEMPORARY ARRAY             0001870
        LDA       MIDHT}            % FIND INDICES FOR TEMPORARY AR-      0001880
                                    %   RAY                               0001890
        LDL(.CNTR).ADFMIDHT}        % ADDRESS OF TEMPORARY ARRAY          0001900
        SLIT(.FLAG)=TEMPPOINTER}                                          0001910
        EXCHL(.FLAG)$ICR}           % JUMP TO TEMPPOINTER                 0001920
        STA       MIDHT}            % STORE UPDATED POINTER              0001930
        LDA       $S}               % TEMPPOINTER PLACED DISTANCE         0001940
                                    %   SQUARED IN RGS                    0001950
% DETERMINE IF STATION IS WITHIN RMAX OF POINT                           0001960
        LDL(.BDC).RUPPSQ}                                                 0001970
        JML       CARBDC}                                                 0001980
        LDA       UPHT}             % INDICES TO TEMPORARY ARRAY         0001990
        LDL(.CNTR).ADFUPHT}         % ADDRESS OF TEMPORARY ARRAY         0002000
        SLIT(.FLAG)=TEMPPOINTER}                                          0002010
        EXCHL(.FLAG)$ICR}           % JUMP TO TEMPPOINTER                 0002020
% DETERMINE RADIUS OF INFLUENCE FOR THIS POINT                           0002030
        LDA       LOWBOTH}          % TOTAL NUMBER HEIGTH REPORTS         0002040
        ADM       LOWHT}            %   IN RLOW                          0002050
        LDL(.BDC) .NLM1}            % NUMBER FOR RLOW MINUS ONE           0002060
        JMG       CARBDC}           % TEST FOR NL OR GREATER              0002070
        SETF      I AND E}          % DISABLE FOR RADIUS STORE PES        0002080
        SETF1     G AND F1}         %   NOT IN RLOW                       0002090
        LDL(.BDC) .RLOW}                                                  0002100
        LDA       CARBDC}                                                 0002110
        STA       RADIUS}                                                 0002120
        SETF      -E AND -F}        % REVERSE MODE                       0002130
        SETF1     -F1 AND -E}                                             0002140
        ADM       MIDBOTH}          % TOTAL NUMBER HEIGTH REPORTS         0002150
        ADM       MIDHT}            %   IN RMID                          0002160
        LDL(.BDC) .NMM1}            % NUMBER FOR RMID MINUS ONE           0002170
        JMG       CARBDC}           % TEST FOR NM OR GREATER              0002180
        SETF      J AND E}          % DISABLE FOR RADIUS STORE PES        0002190
        SETF1     H AND F1}         %   NOT IN RMID                       0002200
        LDL(.BDC) .RMID}                                                  0002210
        LDA       $BDC}                                                   0002220
        STA       RADIUS}                                                 0002230
        SETF      -I AND -E}        % REMAINING PES STORE RMAX            0002240
        SETF1     -H AND -F1}                                             0002250
        LDL(.BDC) .RMAX}                                                  0002260
        LDA       CARBDC}                                                 0002270
        STA       RADIUS}                                                 0002280
% INSERT ALL 0 POINTER INTO STAPERPNT TO MARK START OF HEIGHT ONLY        0002290
        CLRA}                                                             0002300
        LDR       $A                % ALL 0 POINTER                      0002310
        SLIT(.FLAG)=FORMPOINTER}                                          0002320
        EXCAL(.FLAG)$ICR}           % TRANSFER TO FORMPOINTER            0002330
% % INSERT POINTER INDEX TO RGR                                           0002340
% DETERMINE IF STATION IS WITHIN RLOW OF POINT                           0002350
        LDL(.BDC) .RLOWSQ}          % LOWWEST RADIUS OF INFLUENCE         0002360
        JAL       CARBDC}           %   COMPARED WITH DISTANCE           0002370
        SETF      J AND E}          % DISABLE IF STATION OUTSIDE OF       0002380
        SETF1     H AND E}          %   RLOW                              0002390
        SLIT(.FLAG)=FORMPOINTER}                                          0002400
        EXCHL(.FLAG)$ICR}           % TRANSFER                           0002410
```

```
          LDS       $A;                    % SAVE DISTANCE SQUARED          0002420
% UPDATE NUMBER OF REPORTS IN LOW RADIUS                                      0002430
          LDL(.BDC) .PAIR1;                                                  0002440
          LDA       CARBDC;                                                  0002450
          ADM       LOWWND;                                                  0002460
          STA       LOWWND;                                                  0002470
                                                                             0002480
% DETERMINE THE TOTAL NUMBER OF REPORTS WITH BOTH HEIGHT AND WIND            0002490
          SETF      E OR -E;               % ENABLE ALL PES                 0002500
          SETF1     E OR -E;                                                 0002510
          LDA       $R;                    % LOWBOTH                        0002520
          ADM       MIDBOTH;                                                 0002530
          SETF      -I AND E;              % DISABLE FOR RLOW POINTS        0002540
          SETF1     -G AND E1;                                               0002550
          SETF      -J AND E;              % DISABLE FOR RMID POINTS        0002560
          SETF1     -H AND E1;                                               0002570
          ADM       UPBOTH;                                                  0002580
          LDFF1     CARALL;                                                  0002590
          STA       NRBOTH;                                                  0002600
% CALL MAXCOUNT TO DETERMINE MAX FOR ARRAY OF NRBOTH                         0002610
          SLIT(.BDC)=MAXBOTH;            % PASS ARRAY ADDRESS              0002620
          SLIT(.FLAG)=MAXCOUNT;                                            0002630
          EXCHL(.FLAG)$ICR;              % TRANSFER TO MAX COUNT           0002640
% DETERMINE MAXIMUM NUMBER OF REPORTS WITH HEIGHT ONLY                       0002650
          LDA       LOWHT;               % LOWHT                           0002660
          SETF      -I AND E;            % DISABLE FOR RLOW POINTS         0002670
          SETF1     -G AND E1;                                             0002680
          ADM       MIDHT;                                                 0002690
          SETF      -J AND E;            % DISABLE FOR RMID POINTS         0002700
          SETF1     -H AND E1;                                             0002710
          ADM       UPHT;                                                  0002720
          LDFF1     CARALL;              % ENABLE ALL PES                  0002730
          STA       NRHEIGHT;                                              0002740
% CALL MAXCOUNT TO DETERMINE MAX FOR ARRAY OF NRHEIGHT                       0002750
          SLIT(.BDC)=MAXHEIGHT;          % PASS ARRAY ADDRESS              0002760
          SLIT(.FLAG)=MAXCOUNT;                                            0002770
          EXCHL(.FLAG)$ICR;              % TRANSFER TO MAX COUNT           0002780
% DETERMINE MAXIMUM NUMBER OF REPORTS WITH WIND ONLY                         0002790
          LDA       LOWWIND;                                               0002800
          SETF      -I AND E;            % DISABLE FOR RLOW POINTS         0002810
          SETF1     -G AND E1;                                             0002820
          ADM       MIDWIND;                                               0002830
          SETF      -J AND E;            % DISABLE FOR RMID POINTS         0002840
          SETF1     -H AND E1;                                             0002850
          ADM       UPWIND;                                                0002860
          LDFF1     CARALL;                                                0002870
% CALL MAXCOUNT TO DETERMINE MAX FOR ARRAY OF NRWIND                         0002880
          SLIT(.BDC)=MAXWIND;            % PASS ARRAY ADDRESS              0002890
          SLIT(.FLAG)=MAXCOUNT;                                            0002900
          EXCHL(.FLAG)$ICR;              % TRANSFER TO MAX COUNT           0002910
%AFTER ALL POINTS HAVE BEEN CHECKED FOR ALL STATIONS,                       0002920
%RETURN TO CALLER                                                          0002930
          LDL(.FLAG).LINK;                                                 0002940
          STL(.FLAG)$ICR;                                                  0002950
% SUBROUTINE FORMPOINTER BUILDS STAPERPNT BY USE OF POINTERS               0002960
% NEXTSHIFT AND NEXTSTORE. THE NEW POINTER MUST BE IN RGR.                  0002970
% MODE AT ENTRY IS NOT CHANGED.                                            0002980
% ENTRY IS MADE BY EXCHL ON .FLAG. THEREFORE, .FLAG BIT 0 MUST BE RESET.   0002990
P1;       CSR(.FLAG).INNERWD;            % MARK START OF PROCESSING ON     0003000
                                         %   INNER WORD                    0003010
          SETI      E OR -E;             % I BIT SET FOR INNER WORD        0003020
```

```
        SKIP    ,P2;                    % SKIP INITIALIZATION                    0003030
        FILL    2;                      % ALIGN ON EVEN SYLLABLE                 0003040
FORMPOINTER:                                                                    0003050
        STA     T1;                     % SAVE DISTANCE IN T1                    0003060
        LDB     $R;                                                             0003070
        LDB     $D;                     % SAVE MODE PATTERN WITH INDEX           0003080
        LDB     $B;                     %   IN RGB                              0003090
        SETG    F AND E;                % SAVE F BIT IN G BIT                    0003100
        SETI    F AND ~E;               % I BIT RESET FOR OUTER WORD             0003110
P2:     LDA     NEXTSTORE;              % DETERMINE LOCATION FOR POINTER         0003120
        SETF    I AND E;                % ENABLE WHEN PROCESSING INNER           0003130
                                        %   WORD                                0003140
        SWAPA;                          % POINTER TO OUTER WORD                  0003150
        LDF     .ALL;                   % BEGIN PROCESSING OF INDEX IN           0003160
                                        %   OUTER WORD                          0003170
        LDB     $A;                     % SHIFT PART TO RGX                      0003180
        LDX     $B;                                                             0003190
        SHAR    16;                     % FIND INDEX PART                        0003200
        LDS     $A;                     %   AND PLACE IN RGS FOR FETCH          0003210
        LDA     $R;                     % INDEX FOR CURRENT STATION              0003220
        LDL(.BDC) .MASKOUTOTO15;        % MASK OUT MODE BITS                     0003230
        AND     .PDC;                                                           0003240
        RTAL    *-2;                    % SHIFT INTO POINTER POSITION            0003250
                                        %   WITH TWO LOW BITS DROPPED           0003260
        SETF    G AND E;                % RESTORE F BIT                          0003270
        OR      #STAPERPNT;             % CONCATENATE                            0003280
        STA     #STAPERPNT;             % STORE POINTER FOR USE IN ANALY-        0003290
                                        %   SIS ROUTINE                         0003300
        IDF     .ALL;                   % RESUME PROCESSING OF INDEX             0003310
        JXID    =8;                     % COMPUTE NEXT SHIFT PART                0003320
        SETF    J AND E;                % ENABLE ONLY PES COMPLETING             0003330
                                        %   A WORD FOR STAPERPNT                0003340
        LDX   =56;                      % SHIFT PART TO 56                       0003350
        LDF     .ALL;                                                           0003360
        LDB     $X;                     % INTERCHANGE RGX & RGS                  0003370
        LDX     $S;                     %   INDEX PART TO RGX                   0003380
        LDS     $B;                     %   SHIFT PART TO RGS                   0003390
        SETF    J AND E;                                                        0003400
        XI      =1;                     %   AND INCREMENT INDEX PART            0003410
        LDF     .ALL;                   % IN ORDER TO SAVE MODE PATTERN          0003420
        LDB     $X;                     % INDEX PART TO RGB                      0003430
        LDA     $B;                     % AND INTO RGA                           0003440
        SHAL    16;                     %   INTO POINTER WORD POSITION          0003450
        OR      $S;                     % POINTER FORMED IN OUTER WORD           0003460
        SETF    I AND E;                % DISABLE WHEN PROCESSING OUTER          0003470
                                        %   WORD                                0003480
        SWAPA;                          % OUTER WORD DUPLICATED                  0003490
        SETF    G AND E;                % RESTORE F BIT                          0003500
        STA     NEXTSTORE;              % ONLY APPROPRIATE HALF WORD             0003510
                                        %   STORED                              0003520
        CTSRF(.FLAG).INNERWD,FORMPOINTER-1; % SKIP TO DO INNER WORD              0003530
        CRB(.FLAG).INNERWD;             % INNER WORD JUST COMPLETED              0003540
        LDB     $R;                                                             0003550
        LDD     $B;                     % RESTORE MODE PATTERN                   0003560
        LDA     T1;                     % RESTORE RGA                            0003570
        LDL(.FLAG)$ICR;                 % RETURN TO CALLING SEQUENCE             0003580
% SUBROUTINE BUILDS FULL WORD POINTERS TO TEMPORARY ARRAYS.                      0003590
        FILL    2;                                                              0003600
TEMPPOINTER:                                                                    0003610
        LDB     $R;                                                             0003620
        LDB     $D;                     % SAVE MODE BITS                        0003630
```

```
        LDR       $B;
        LDB       $A;                  % OUTER INDEX TO RGX
        LDX       $B;
        LDEE1     .ALL;                % ENABLE POINTER FETCH AND SWAPA
        SWAPA;                         % INNER INDEX TO OUTER WORD
        LDB       $S;                  % DISTANCE SQUARED OUT OF RGS
        LDS       $A;                  % INNER INDEX TO RGS
        LDA       $B;
        LDB       $R;                  % RESTORE MODE BITS
        LDD       $B;
        SETE1     E1 AND -E1;          % DISABLE INNER WORD
        STR       *0(.CNTR);           % STORE OUTER WORD
        LDB       $R;
        LDD       $B;
        SETE      E AND -E;            % DISABLE OUTER WORD
        STR       *0(.CNTR);           % STORE INNER WROD
        LDEE1     .ALL;
        LDB       $S;                  % INNER INDEX TO RGA
        LDS       $A;                  % DISTANCE SQUARED TO RGS
        LDA       $B;
        SWAPA;                         % INNER INDEX BACK TO INNER WORD
        SETE1     E AND -E;            % DISABLE INNER WORD
        LDB       $X;                  % OUTER INDEX BACK
        LDA       $B;                  %   INTO RGA OUTER WORD
        LDB       $R;                  % RESTORE MODE
        LDD       $B;
        LDL(.BDC) .PAIR1;
        ADM       .BDC;                % INCREMENT FOR STORE
        STL(.FLAG)$ICR;                % RETURN
% STORAGE LOCATIONS
MAXSTAPERPE:        EQU  220;          % MAXIMUM STAIONS USED IN ANY PE
POINTORDER:    BLK  2;                 % EACH HALF BYTE POINTS TO A GRID
                                       %   POINT

ICURRENT:      BLK  1;
JCURRENT:      BLK  1;                 % FOR TEMPORARY STORAGE
NEXTSTORE:     BLK  1;                 % POINTER TO STAPERPNT
NEXTSHIFT:     BLK  1;
NOBOTH:        BLK  1;
NUMBERLOW:     BLK  1;
MIDPOINTER:    BLK  1;                 % POINTER TO MIDSTATONS
UPPOINTER:     BLK  1;                 % POINTER TO UPSTATIONS
STAPERPNT:     BLK  128;               % POINTERS TO STATIONS TO BE USED
                                       %   IN ANALYSIS AT EACH POINT

FIELD:         BLK  84;                % ARRAY FOR STORAGE OF FIELD
                                       %   BEING ANALYZED

T1:            BLK  1;                 % TEMPORARY STORAGE LOCATIONS
T2:            BLK  1;
T3:            BLK  1;
ISTART:        BLK  1;                 % THE LEFT-MOST COLUMN IN BLOCK
JSTART:        BLK  1;                 % THE LOWER ROW IN BLOCK
MIDSTATIONS:   BLK  15;                % FOR TEMPORARY STORAGE WHILE
UPSTATIONS:    BLK  15;                %   DETERMINING STATIONS USED
NRBOTH:        BLK  1;                 % NUMBER OF REPORTS WITH BOTH
                                       %   HEIGHT AND WIND

NRHT:          BLK  1;                 % NUMBER OF REPORTS OF HEIGHT
                                       %   ONLY
NRWND:        BLK  1;                 % NUMBER OF REPORTS WITH ONLY
                                       %   WIND

               FILL 256;               % STADATA MUST BE ALIGNED ON FOUR
                                       %   ROW BOUNDARY--LAST TWO BITS
                                       %   DROPPED FOR STAPOINTER
```

```
STADATA:      BLK  4*MAXSTAPERPE;      * STORAGE FOR DATA TO BE USED IN 0004250
                                       *   EACH PE FOR ANALYSIS         0004260
* CU SYMBOL DEFINITIONS                                                 0004270
.BDC:              EQU  $C0;                                           0004280
.ALL:              EQU  $C1;                                           0004290
.CNTR:             EQU  $C2;                                           0004300
.FLAG:             EQU  $C3;           * CU FLAG WORD                  0004310
.STEP4UNTIL256:    EQU  $D0;                                           0004320
.MASKOUTOTO28:     EQU  $D1;                                           0004330
.MASKOUTOTO15:     EQU  $D2;                                           0004340
.STEP4UNTIL0:      EQU  $D3;                                           0004350
.PAIR8:            EQU  $D4;                                           0004360
.SLOWSQ:           EQU  $D5;                                           0004370
.SMIDSQ:           EQU  $D6;                                           0004380
.PAIR1:            EQU  $D7;                                           0004390
.SUPERSQ:          EQU  $D8;                                           0004400
.PAIR5:            EQU  $D9;                                           0004410
.ADFUP:            EQU  $D10;                                          0004420
.ADFMID:           EQU  $D11;                                          0004430
.MAXNORUP:         EQU  $D62;                                          0004440
.STORESTEP4:       EQU  $D59;                                          0004450
.TEMP:             EQU  $D60;                                          0004460
.CNTRSTORE:        EQU  $D61;                                          0004470
.MAXNORMID:        EQU  $D63;                                          0004480
END.                                                                   0004490
```

```
 ROUTINE FIELDSCAN
BEGIN                                                                   0000010
% SCANFIELD PERFORMS THE ADJUSTMENT OF THE FIELD                        0000020
% DEFINE STATEMENTS FOR CU REGISTERS                                    0000030
                    DEFINE CARBDC=$C0##;                                0000040
                    DEFINE CARALL=$C1##;                                0000050
                    DEFINE CARCNTR=$C2##;                               0000060
SCANFIELD:     FILL 2;                  % ONTO A WORD BOUNDARY          0000070
          SLIT(.BDC)=MAXBOTHSTA;                                        0000080
          LOAD(.BDC),MAXCYCLE;          % START FETCH OF MAX CYCLES/POINT0000090
% INITIALIZE FOR FIRST POINT                                            0000100
          LDL(.CNTR),STEP1UNTIL15;                                      0000110
          LDA       POINTORDER;                                         0000120
          SLIT(.BDC)=28;                % SHIFT TO UNPACK FIRST POINT   0000130
% FIND POINTER TO NEXT POINT                                            0000140
SF1:      SHAR      0(.BDC);                                            0000150
          ALIT(.BDC)=-4;                                                0000160
          LDX       CARCNTR;            % RADII STORED IN SAME ORDER    0000170
          LDA       *RADII;             %   AS POINTS ARE USED        0000180
          STA       RADIUS;                                           0000190
          STL(.BDC) ,PTSHIFT;                                           0000200
% FIND INDEX TO FIELD FOR THIS POINT                                    0000210
          LDS       $A;                                               0000220
          LDL(.BDC) ,PAIR8;                                             0000230
          IML        CARBDC;             % TEST FOR ROW OF POINT      0000240
          SETE      I AND E;                                          0000250
          SETE1     G AND E1;                                         0000260
          LDL(.BDC) ,PAIR28;                                            0000270
          ADM       CARBDC;             % INDEX FOR ROW 1 POINTS       0000280
          SETE      -E AND -E;          % FOR POINTS ON SECOND ROW      0000290
          SETE1     -E1 AND -E1;                                      0000300
          LDL(.BDC) ,PAIR33;                                            0000310
          ADM       CARBDC;                                           0000320
          SETE      E OR -E;                                          0000330
          SETE1     E1 OR -E1;                                        0000340
          STA       FIELDINDEX;                                       0000350
          LDA       $R;                                               0000360
          LDEF1     CARALL;                                           0000370
% FIND FIRST GUESS VALUES                                              0000380
          LDB        $A;                                              0000390
          LDX       $B;                                               0000400
          SWAPA;                        % INNER INDEX TO OUTER WORD    0000410
          LDS       $A;                                               0000420
          LDS       $B;                 % AND TO RGS                   0000430
          LDA       *FIELD;             % FETCH OUTER WORD             0000440
          SETE      E AND -E;           % AND PROTECT                  0000450
          LDA       #FIELD;             % FETCH INNER WORD             0000460
          SETE      I OR E;             % RESTORE MODE                 0000470
          STA       GPFIELD;            % TEMPORARY STORAGE            0000480
          LDB        $A;                                              0000490
          LDX       $B;                                               0000500
          SWAPA;                        % INNER INDEX TO OUTER WORD    0000510
          LDS       $A;                                               0000520
          LDS       $B;                 % AND TO RGS                   0000530
          LDA       *TEMP;              % FETCH OUTER WORD             0000540
          SETE      E AND -E;           % AND PROTECT                  0000550
          LDA       #TEMP;              % FETCH INNER WORD             0000560
          SETE      I OR E;             % RESTORE MODE                 0000570
          STA       GPTEMP;             % TEMPORARY STORAGE            0000580
```

```
% INITIALIZE TEMPORARY STORAGE LOCATIONS                                0000590
        CLRA;                                                          0000600
        STA     NEXTSTORE;                                             0000610
        STA     WTSUMBOTH;                                             0000620
        STA     WTSUMWND;                                              0000630
        STA     WTSUMHT;                                               0000640
        STA     WTSUMTEMP;                                             0000650
        STA     NRHT;                                                  0000660
        STA     NRBOTH;                                                0000670
        STA     NRWND;                                                 0000680
        STA     NRTEMP;                                                0000690
% FORM COUNTER FOR STATIONS OF TYPE BOTH                               0000700
        SLIT(.CNTR)=1;                  % INCREMENT                   0000710
        CSHL(.CNTR)24;                                                 0000720
        CADD(.CNTR),MAXCYCLE;           % LIMIT                       0000730
        CSHL(.CNTR)24;                                                 0000740
        SLIT(.CNTR)=1;                  % STARTING VALUE              0000750
% COMPLETE PFM INITIALIZATION FOR SCAN 4                               0000760
        CTSBF(.FLAG).SCAN4,SF2;         % SKIP UNLESS SCAN 4          0000770
        STA     WTSUMBOTH;                                             0000780
        STA     WTSUMWND;                                              0000790
% AND FOR SCANS 2,3,4                                                  0000800
% INITIALIZE NEXTSHIFT                                                 0000810
        LDL(.BDC) .PAIR24;                                             0000820
        LDA     CARBDC;                                                0000830
        STA     NEXTSHIFT;                                             0000840
% COMPUTE CORRECTION FOR THIS STATION                                  0000850
SF2:    SLIT(.FLAG)=UNPACK;                                            0000860
        EXCHL(.FLAG)$ICR;               % TRANSFER TO UNPACK          0000870
        CTSBT(.FLAG).SCAN1,SF4;         % SKIP FOR SCAN 1             0000880
        LDA     UDFWIND;                                               0000890
        LDL(.BDC) .PAIR512SCALE;        % 512:SCALE 1/16              0000900
        SBM     CARBDC;                 % CONVERT DISPLACEMENT FORM  TO 0000910
                                        %   SIGN/MAGNITUDE FORM        0000920
        MLM     DELTAI;                 % I COMPONENT OF CHANGE:SCALE  0000930
                                        %   SCALE 1/256                0000940
        LOS     $A;                                                    0000950
        LDA     UDFWIND;                                               0000960
        SBM     CARBDC;                 % TO SIGN/MAGNITUDE FORM      0000970
        MLM     DELTAJ;                                                0000980
        ADM     $S;                     % GADIENT CHANGE:SCALE 1/256  0000990
        SHAR    4;                      % SCALE TO 1/16               0001000
        ADM     HEIGHT;                                                0001010
        SBM     GPFIELD;                % CORRECTON COMPUTED          0001020
        MLM     WEIGHT;                 % WEIGHTED CORRECTION:SCALE 1/2560001030
        SHAR    4;                      % SCALE TO 1/16               0001040
        ADM     $S;                     % CORRECTION CALCULATED       0001050
        ADM     WTCORBOTH;                                             0001060
        STA     WTCORBOTH;                                             0001070
% INCREMENT COUNTERS FOR DENOMINATOR                                   0001080
        CTSBF(.FLAG).SCAN4,SFI1;        % SKIP UNLESS SCAN 4          0001090
        LDA     WTSUMBOTH;                                             0001100
        ADM     WEIGHT;                                                0001110
        STA     WTSUMBOTH;                                             0001120
SFI1:   LDL(.BDC) .PAIR1;                                              0001130
        LDA     NRBOTH;                                                0001140
        ADM     CARBDC;                                                0001150
        STA     NRBOTH;                                                0001160
% NOW COMPUTE CORRECTION ON BASIS OF HEIGHT ALONE                      0001170
SF4:    LDA     HEIGHT;                                                0001180
        SBM     AVEHEIGHT;                                             0001190
```

```
         MLM       WEIGHT;          % WEIGHTED CORRECTION;SCALE 1/2560001200
         SHAR      4;               % SCALE TO 1/16                  0001210
         ADM       WTCORHT;                                         0001220
         STA       WTCORHT;                                         0001230
% INCREMENT COUNTERS FOR DENOMINATOR                                 0001240
         CTSBF(.FLAG),SCAN4,SFI2;   % SKIP UNLESS SCAN 4            0001250
         LDA       WTSUMHT;                                         0001260
         ADM       WEIGHT;                                          0001270
         STA       WTSUMHT;                                         0001280
SFI2;    LDL(.BDC) .PAIR1;                                          0001290
         LDA       NRHT;                                            0001300
         ADM       CARBDC;                                          0001310
         STA       NRHT;                                            0001320
% CORRCTION FOR TEMPERATURE                                         0001330
         CTSBT(.FLAG),TEMPANAL,SF5; % SKIP IF NO TEMP ANALYSIS      0001340
         LDA       TEMPERATURE;                                     0001350
         SBM       AVETEMP;                                         0001360
         MLM       WEIGHT;          % WEIGHTED CORRECTION;SCALE 1/2560001370
         SHAR      4;               % SCALE TO 1/16                  0001380
         ADM       WTCORTEMP;                                       0001390
         STA       WTCORTEMP;                                       0001400
% INCREMENT COUNTERS FOR DENOMINATOR                                 0001410
         CTSBF(.FLAG),SCAN4,SFI3;   % SKIP UNLESS SCAN 4            0001420
         LDA       WTSUMHT;                                         0001430
         ADM       WEIGHT;                                          0001440
         STA       WTSUMHT;                                         0001450
SFI3;    LDL(.BDC) .PAIR1;                                          0001460
         LDA       NRTEMP;                                          0001470
         ADM       CARBDC;                                          0001480
         STA       NRTEMP;                                          0001490
SF5;     TXLTM(.CNTR),SF2;          % NEXT STATION                 0001500
         SLIT(.BDC)=MAXHTSTA;                                       0001510
         LOAD(.BDC),MAXCYCLE;       % START FETCH OF MAX CYCLES/POINT0001520
         LDA       NEXTSHIFT;                                       0001530
         LDL(.BDC) .PAIR8;                                          0001540
         SBM       CARBDC;          % DECREMENT SHIFT COUNT         0001550
         JML       =0;              % TEST FOR COUNT LESS THAN 0    0001560
         SETE      J AND E;         % ENABLE IF SO                  0001570
         SETE1     H AND E;                                         0001580
         LDL(.BDC) .PAIR24;                                         0001590
         LDA       CARBDC;          % RESET SHIFT COUNTERS          0001600
         SETF      F OR -E;         % ENABLE FOR STORE              0001610
         SETE1     E OR -E;                                         0001620
         SETE      J AND E;         % ENABLE IF NEW ROW             0001630
         SETE1     H AND E1;                                        0001640
         LDA       NEXTSTORE;       % FOR UPDATE                   0001650
         LDL(.BDC) .PAIR1;                                          0001660
         ADM       CARBDC;                                          0001670
         STA       NEXTSTORE;       % UPDATED POINTER              0001680
         SETE      F OR -E;                                         0001690
         SETE1     F OR -E;                                         0001700
% FORM COUNTER FOR HEIGHT ONLY STATIONS                             0001710
         SLIT(.CNTR)=1;             % INCREMENT                     0001720
         CSHL(.CNTR)24;                                           0001730
         CADD(.CNTR),MAXCYCLE;      % LIMIT                         0001740
         CSHL(.CNTR)24;                                           0001750
         SLIT(.CNTR)=1;             % STARTING VALUE                0001760
% COMPUTE HEIGHT ONLY CORRECTIONS                                  0001770
SF6;     LDA       HEIGHT;                                          0001780
         SBM       AVEHEIGHT;                                       0001790
         MLM       WEIGHT;          % WEIGHTED CORRECTION;SCALE 1/2560001800
```

```
        SHAR    4;                  * SCALE TO 1/16                     0001810
        ADM     WTCORHT;                                                0001820
        STA     WTCORHT;                                                0001830
* INCREMENT COUNTERS FOR DENOMINATOR                                    0001840
        CTSPF(.FLAG),SCAN4,SF14;    * SKIP UNLESS SCAN 4                0001850
        LDA     WTSUMHT;                                                0001860
        ADM     WEIGHT;                                                 0001870
        STA     WTSUMHT;                                                0001880
SF14;   LDL(.BDC) .PAIR1;                                               0001890
        LDA     NRHT;                                                   0001900
        ADM     CARBDC;                                                 0001910
        STA     NRHT;                                                   0001920
* CORRCTION FOR TEMPERATURE                                             0001930
        CTSRT(.FLAG),TEMPANAL,SF5;  * SKIP IF NO TEMP ANALYSIS          0001940
        LDA     TEMPERATURE;                                            0001950
        SBM     AVETEMP;                                                0001960
        MLM     WEIGHT;             * WEIGHTED CORRECTION;SCALE 1/2560  0001970
        SHAR    4;                  * SCALE TO 1/16                     0001980
        ADM     WTCORTEMP;                                              0001990
        STA     WTCORTEMP;                                              0002000
* INCREMENT COUNTERS FOR DENOMINATOR                                    0002010
        CTSRF(.FLAG),SCAN4,SF15;    * SKIP UNLESS SCAN 4                0002020
        LDA     WTSUMHT;                                                0002030
        ADM     WEIGHT;                                                 0002040
        STA     WTSUMHT;                                                0002050
SF15;   LDL(.BDC) .PAIR1;                                               0002060
        LDA     NRTEMP;                                                 0002070
        ADM     CARBDC;                                                 0002080
        STA     NRTEMP;                                                 0002090
SF7;    TXLTM(.CNTR),SF6;           * NEXT STATION                      0002100
* SKIP WIND ONLY ON SCAN 1                                              0002110
        CTSRT(.FLAG),SCAN1,COMPUTE;                                     0002120
        SLIT(.BDC)=MAXWNDSTA;                                           0002130
        LOAD(.BDC),MAXCYCLE;        * START FETCH OF MAX CYCLES/POINT   0002140
* MOVE POINTER PAST DELIMETER                                           0002150
        LDA     NEXTSHIFT;                                              0002160
        LDL(.BDC) .PAIR8;                                               0002170
        SBM     CARBDC;             * DECREMENT SHIFT COUNT             0002180
        JML     =0;                 * TEST FOR COUNT LESS THAN 0        0002190
        SETF    J AND E;            * ENABLE IF SO                      0002200
        SETF1   H AND E;                                                0002210
        LDL(.BDC) .PAIR24;                                              0002220
        LDA     CARBDC;             * RESET SHIFT COUNTERS              0002230
        SETE    F OR -E;            * ENABLE FOR STORE                  0002240
        SETE1   F OR -E;                                                0002250
        SETE    J AND E;            * ENABLE IF NEW ROW                 0002260
        SETF1   H AND E1;                                               0002270
        LDA     NEXTSTORE;          * FOR UPDATE                        0002280
        LDL(.BDC) .PAIR1;                                               0002290
        ADM     CARBDC;                                                 0002300
        STA     NEXTSTORE;          * UPDATED POINTER                   0002310
        SETE    F OR -E;                                                0002320
        SETE1   F OR -E;                                                0002330
* FORM COUNTER FOR HEIGHT ONLY STATIONS                                 0002340
        SLIT(.CNTR)=1;              * INCREMENT                         0002350
        CSHL(.CNTR)24;                                                  0002360
        CADD(.CNTR),MAXCYCLE;       * LIMIT                             0002370
        CSHL(.CNTR)24;                                                  0002380
        SLIT(.CNTR)=1;              * STARTING VALUE                    0002390
SF8;    SLIT(.FLAG)=UNPACK;                                             0002400
        EXCHL(.FLAG)$ICR;           * TRANSFER TO UNPACK                0002410
```

```
        LDA       UOFWIND;
        LDL(.BDC) .PAIR512SCALE;      % 512:SCALE 1/16
        SBM       CARBDC;             % CONVERT DISPLACEMENT FORM  TO
                                      %   SIGN/MAGNITUDE FORM
        MLM       DELTAI;             % I COMPONENT OF CHANGE:SCALE
                                      %   SCALE 1/256
        LDS       $A;
        LDA       VOFWIND;
        SBM       CARBDC;             % TO SIGN/MAGNITUDE FORM
        MLM       DELTAJ;
        ADM       $S;                 % GADIENT CHANGE:SCALE 1/256
        SHAR      4;                  % SCALE TO 1/16
        ADM       AVEHEIGHT;
        SBM       GPFIELD;            % CORRECTON COMPUTED
        MLM       WEIGHT;             % WEIGHTED CORRECTION:SCALE 1/256
        SHAR      4;                  % SCALE TO 1/16
        ADM       $S;                 % CORRECTION CALCULATED
        ADM       WTCORWND;
        STA       WTCORWND;
% INCREMENT COUNTERS FOR DENOMINATOR
        CTSBF(.FLAG).SCAN4,SFI7;      % SKIP UNLESS SCAN 4
        LDA       WTSUMWND;
        ADM       WEIGHT;
        STA       WTSUMWND;
SFI7:   LDL(.BDC) .PAIR1;
        LDA       NRWND;
        ADM       CARBDC;
        STA       NRWND;
        TXLTM(.CNTR),SF8;
% COMPUTE TOTAL CORRECTION
        LDA       WTCORHT;
        SHAR      3;                  % MULTIPLY BY A=1/8
        LDS       $A;
        CTSBT(.FLAG).SCAN1,FINDDENOM;
        LDA       WTCORWND;
        LDL(.BDC) .BCONS;
        SHAR      0(.BDC);            % MULTIPLY BY B=1/8,1/4,1/2
        ADM       $S;
        ADM       WTCORBOTH;
FINDDENOM:
        LDR       $A;
        CTSBF(.FLAG).SCAN4,DENOM2;
        LDA       NRBOTH;
        ADM       NRHT;
        ADM       NRWND;
        LDL(.BDC) .PAIR2;
        IMG       CARBDC;
        SETE      I AND E;            % FOR SPECIAL SCAN 4 DENOMINATOR
        SETE1     G AND E1;
        LDA       WTSUMHT;
        SHAR      2;                  % MULTIPLY BY 1/4
        ADM       WTSUMWND;
        SHAR      1;                  %          BY 1/2
        ADM       WTCORBOTH;
        LDS       $A;
        SET       -E AND -E;          % FOR DENOM2
        SET       -E1 AND -E1;
DENOM2: LDA       NRHT;
        SHAR      3;
        LDS       $A;
        LDA       NRWND;
```

```
        SHAR    0(,BDC)$                                                0003030
        ADM     NRBOTH$                                                 0003040
        ADM     $S$                                                     0003050
        LDS     $A$                                                     0003060
        LDA     $R$             % NUMERATOR                             0003070
        SHAR    A$              % SCALE FOR DIVISION                    0003080
        SETE    F OR -E$                                                0003090
        SET     E OR -E$                                                0003100
        SHAL    8$              % IN ORDER TO BE SAME SCALE AS          0003110
        ILM$                    % NUMERATOR IS ZERO WHEN DENOMI-        0003120
        SETE    -I AND E$       %   NATOR IS                            0003130
        SETE1   -G AND E1$                                              0003140
        DVRM    $S$             %   NUMERATOR AFTER DIVIDE              0003150
        SHAL    8$              % SCALE                                 0003160
        ADM     FIELD$                                                  0003170
        STA     FIELD$                                                  0003180
% FOR TEMPERATURE                                                       0003190
        LDA     WTCORTEMP$                                              0003200
        ILM$                    % NUMERATOR IS ZERO WHEN DENOMI-        0003210
        SETE1   -G AND E1$                                              0003220
        SETE    -I AND E$       %   NATOR IS                            0003230
        DVRM    NRTEMP$                                                 0003240
        SHAL    8$              % SCALE                                 0003250
        ADM     TEMPFIELD$                                              0003260
        STA     TEMPFIELD$                                              0003270
        TXLTH(.CNTR),SF1$       % NEXT POINT                            0003280
        HALT$                                                           0003290
% UNPACK TAKES NEXTSTORE AND NEXTSHIFT, EXTRACTS THE NEXT STAPERPNT      0003300
% POINTER TO STADATA, AND USES THIS POINTER TO FETCH                     0003310
% THE 1ST, 3RD, AND 4TH WORDS OF THE STATION REPORT, UNPACKS U,V COMP/-  0003320
% NENTS OF WIND, HEIGHT, TEMPERATURE(IF .TEMPANAL IS SET), THE I AND J   0003330
% STATION COORDINATES ARE OBTAINED, AND THE WEIGHTING FACTOR:            0003340
%       W=(DISTANCE**2-RADIUS**2)/(DISTANCE**2+RADIUS**2                0003350
% IS COMPUTED AND RETURNED IN RGS. THE I,J COMPONENTS OF DISTANCE ARE    0003360
% STORED IN DELTAI AND DELTA J, RESPECTIVELY.                            0003370
%                                                                       0003380
% THE POINTER FROM STAPERPNT IS TESTED FOR 0 AND THE RESULT IS ANDED     0003390
% INTO BOTH I(JG) AND E(E1) BITS                                         0003400
UNPACK$ FILL    2$              % ALIGN ON WORD BOUNDARY                 0003410
% FIND ROW ADDRESS IN STAPERPNT                                         0003420
        SETE    F OR -E$                                                0003430
        SETE1   E OR -E$                                                0003440
        LDA     NEXTSTORE$                                              0003450
        LDR     $A$                                                     0003460
        LDX     $B$             % OUTER INDEX TO RGX                    0003470
        SWAPA$                                                          0003480
        LDS     $A$             % INNER INDEX TO RGS                    0003490
        LDA     *STAPERPNT$     % FETCH OUTER WORD                      0003500
        SETE    F AND -E$       %   AND DISABLE                         0003510
        LDA     $STAPERPNT$     % FETCH INNER WORD                      0003520
        SETE    F OR -E$        % ENABLE OUTER WORD                      0003530
        LDR     $A$             % SAVE STAPERNT WORD                    0003540
% FIND BYTE WITHIN WORD                                                 0003550
        LDA     NEXTSHIFT$                                              0003560
        LDR     $A$             % OUTER COUNT TO RGX                    0003570
        LDX     $9                                                      0003580
        SWAPA$                  % INNER COUNT TO RGS                    0003590
        LDS     $A$                                                     0003600
        SWAPA$                                                          0003610
        LDL(.BDC) .PAIR8$                                               0003620
        SBM     CARBDC$         % DECREMENT SHIFT COUNT                 0003630
```

```
        JML         =0;                 % TEST FOR COUNT LESS THAN 0      0003640
        SETF        J AND E;            % ENABLE IF SO                    0003650
        SETF1       H AND E;                                             0003660
        LDL(.BDC)   .PAIR24;                                             0003670
        LDA         CARBDC;             % RESET SHIFT COUNTERS           0003680
        SETF        F OR -E;            % ENABLE FOR STORE                0003690
        SETF1       E OR -E;                                             0003700
        SETF        J AND F;            % ENABLE IF NEW ROW               0003710
        SETF1       H AND E1;                                            0003720
        LDA         NEXTSTOREF;         % FOR UPDATE                     0003730
        SETF        E AND E;            % DISABLE WHEN COMPLETED DATA TYPE0003740
        SETF1       G AND E1;                                            0003750
        LDL(.BDC)   .PAIR1;                                              0003760
        ADM         CARBDC;                                              0003770
        STA         NEXTSTOREF;         % UPDATED POINTER                0003780
        LDA         $R;                 % STAPERPNT WORD                 0003790
        SETF1       F1 AND -E1;         % DISABLE                        0003800
        RTAR        *-2;                % OUTER INDEX INTO POSITION       0003810
        SETF        E AND -;                                             0003820
        SETF        E AND -F;           % DISABLE OUTER WORD              0003830
        SETF1       F1 OR-E1;           % ENABLE INNER WORD               0003840
        RTAR        *-2;                % INNER INDEX INTO POSITION       0003850
        LDL(.BDC)   .MASKOUTOT021ANDLAST2;                               0003860
        AND         CARBDC;             % INDICES TO STATIONS NOW FORMED  0003870
% TEST FOR ALL ZERO POINTER                                              0003880
        JLZ;                                                             0003890
        SETF        -J AND E;           % DISABLE IF DELIMETER            0003900
        SETF1       -H AND E1;                                           0003910
        SETI        E OR E;             % UPDATE I AND G                 0003920
        SETG        E1 OR E1;                                            0003930
        LDB         $A;                 % OUTER POINTER TO RGX           0003940
        LDX         $B;                                                  0003950
        SETF        F OR -E;            % ENABLEOUTER WORD FOR SWAPA      0003960
        SWAPA;                          % INNER POINTER TO OUTER WORD     0003970
        LDS         $A;                 %   OF RGS                       0003980
% FETCH U,V COMPONENTS OF WIND AND PUT IN TEMPORARY STORAGE               0003990
        LDA         *STADATA-1;         % FETCH OUTER WORD; RGX POINTS +40004000
        SETF        F AND -E;           % PROTECT OUTER WORD              0004010
        LDA         #STADATA-1;         % FETCH INNER WORD                0004020
        SETF        I OR E;             % RESTORE E BIT                   0004030
        LDR         $A;                 % SAVE WIND WORD                  0004040
        LDL(.BDC)   .MASKOUT015;                                         0004050
        AND         .BDC                % V COMPONENT;SCAL 1/16           0004060
        STA         VDFWIND;                                             0004070
        LDA         $R;                                                  0004080
        SHAR        16;                 % U COMPONENT;SCALE 1/16         0004090
        STA         UDFWIND;                                             0004100
% FETCH HEIGHT AND TEMPERATURE AND PUT IN TEMPORARY STORAGE              0004110
        LDA         *STADATA-2;         % FETCH OUTER WORD               0004120
        SETF        F AND -E;           %   AND PROTECT                  0004130
        LDA         #STADATA-2;         % FETCH INNER WORD               0004140
        SETF        I OR E;             % RESTORE E BIT                   0004150
        LDR         $A;                                                  0004160
        AND         CARBDC;             % HEIGHT;SCALE 1/16               0004170
        STA         HEIGHT;                                              0004180
        LDA         $R;                                                  0004190
        SHAR        16;                                                  0004200
        STA         TEMPERATURE;                                         0004210
% FETCH INTERPOLATED HEIGHT AND TEMP VALUES                              0004220
        LDA         *STADATA-3;         % FETCH OUTER WORD               0004230
        SETF        F AND -E;           %   AND PROTECT                  0004240
```

```
          LDA       *STADATA-3;          * FETCH INNER WORD                    0004250
          SETF      I OR F;              * RESTORE F BIT                       0004260
          LDR       $A;                                                        0004270
          AND       CARBDC;              * HEIGHT:SCALE 1/16                   0004280
          STA       HEIGHT;                                                    0004290
          LDA       $R;                                                        0004300
          SHAR      16;                                                        0004310
          STA       TEMPERATURE;                                               0004320
* FETCH STATION COORDINATES AND COMPUTE I,J COMPONENTS OF DISTANCE            0004330
UP1:      LDA       *STADATA-4;          * FETCH OUTER WORD                    0004340
          SETF      F AND -E,            *   AND PROTECT                       0004350
          LDA       *STADATA-4;          * FETCH INNER WORD                    0004360
          SETF      I OR F;              * RESTORE F                           0004370
          LDR       $A;                  * SAVE                                0004380
          AND       CARBDC;              * JSTATION:SCALE 1/16                 0004390
          SBM       JCURRENT;            * JSTATION-JPOINT:SCALE 1/16          0004400
          LDS       $A;                  * TO RGS                              0004410
          LDA       $R;                                                        0004420
          SHAL      16;                  * ISTATION:SCALE 1/16                 0004430
          SBM       ICURRENT;            * ISTATION-IPOINT:SCALE 1/16          0004440
          CTSRT(.FLAG),HTONLY,UP2;       * SKIP STORE IF HEIGHT ONLY           0004450
          STA       DELTAI;                                                    0004460
          STS       DELTAJ;                                                    0004470
UP2:      MLM       $A;                  * DELTAI*2:SCALE 1/256                0004480
          LDR       $S;                                                        0004490
          LDS       $A;                  * DELTAI*2                            0004500
          LDA       $B                   * DELTAJ                              0004510
          MLM       $R;                  * DELTAJ*2:SCALE 1/256                0004520
          ADM       $S;                  * DISTANCE SQURRED:SCALE 1/256        0004530
          LDS       $A;                  * DISTANCE SQUARED FOR W              0004540
          LDA       RADIUS;                                                    0004550
          LDR       $A;                  * RADIUS SQUARED                      0004560
          ADM       $S;                  * R*2+D*2                             0004570
          LDR       $A                   * R*2+D*2                             0004580
          LDA       $R;                  * RADIUS SQUARED                      0004590
          LDR       $R;                  * R*2+D*2                             0004600
          SBM       $S;                  * R*2-D*2                             0004610
          SHAL      8;                   * SCALE FOR DIVISION                  0004620
          DVM       $R                   * (R*2-D*2)/(R*2+D*2)=W:              0004630
          STA       WEIGHT;              *   SCALE 1/256                       0004640
SF :      STL(.FLAG)$ICR                 * RETURN                              0004650
* STORAGE LOCATIONS                                                           0004660
MAXSTAPERPE:        EQU  220;            * MAXIMUM STAIONS USED IN ANY PE      0004670
POINTORDER:   BLK   2;                   * EACH HALF BYTE POINTS TO A GRID     0004680
                                         *   POINT                             0004690
ICURRENT:     BLK   1;                                                        0004700
JCURRENT:     BLK   1;                   * FOR TEMPORARY STORAGE               0004710
NEXTSTORE:    BLK   1;                   * POINTER TO STAPERPNT                0004720
NEXTSHIFT:    BLK   1;                                                        0004730
NOROTH:       BLK   1;                                                        0004740
NUMBERLOW:    BLK   1;                                                        0004750
MIDPOINTER:   BLK   1;                   * POINTER TO MIDSTATONS               0004760
UPPOINTER:    BLK   1;                   * POINTER TO UPSTATIONS               0004770
STAPERPNT:    BLK   128;                 * POINTERS TO STATIONS TO BE USED     0004780
                                         *   IN ANALYSIS AT EACH POINT         0004790
FIELD:        BLK   84;                  * ARRAY FOR STORAGE OF FIELD          0004800
                                         *   BEING ANALYZED                    0004810
T1:           BLK   1;                   * TEMPORARY STORAGE LOCATIONS         0004820
T2:           BLK   1;                                                        0004830
T3:           BLK   1;                                                        0004840
ISTART:       BLK   1;                   * THE LEFT-MOST COLUMN IN BLOCK       0004850
```

```
JSTART:          BLK   1;
MIDSTATIONS:     BLK   15;
UPSTATIONS:      BLK   15;
NRBOTH:          BLK   1;

NRHT:            BLK   1;

NRWND:           BLK   1;

                 FILL  256;


STADATA:         BLK   4xMAXSTAPERPE;

% CU SYMBOL DEFINITIONS
.BDC:                  EQU   $C0;
.ALL:                  EQU   $C1;
.CNTR:                 EQU   $C2;
.FLAG:                 EQU   $C3;
.STEP4UNTIL256:        EQU   $D0;
.MASKOUTOTO28:         EQU   $D1;
.MASKOUTOTO15:         EQU   $D2;
.STEPM4UNTIL0:         EQU   $D3;
.PAIR8:                EQU   $D4;
.RLOWSQ:               EQU   $D5;
.RMIDSQ:               EQU   $D6;
.PAIR1:                EQU   $D7;
.RUPPSQ:               EQU   $D8;
.PAIR5:                EQU   $D9;
.ADFUP:                EQU   $D10;
.ADFMID:               EQU   $D11;
.MAXNORUP:             EQU   $D62;
.STORESTEPM4:          EQU   $D59;
.TEMP:                 EQU   $D60;
.CNTRSTORE:            EQU   $D61;
.MAXNORMID:            EQU   $D63;
END.
```

```
% THE LOWER ROW IN BLOCK                0004860
% FOR TEMPORARY STORAGE WHILE           0004870
%   DETERMINING STATIONS USED           0004880
% NUMBER OF REPORTS WITH BOTH           0004890
%   HEIGHT AND WIND                     0004900
% NUMBER OF REPORTS OF HEIGHT           0004910
%   ONLY                                0004920
% NUMBER OF REPORTS WITH ONLY           0004930
%   WIND                                0004940
% STADATA MUST BE ALIGNED ON FOUR       0004950
%   ROW BOUNDARY--LAST TWO BITS         0004960
%   DROPPED FOR STAPOINTER              0004970
% STORAGE FOR DATA TO BE USED IN        0004980
%   EACH PE FOR ANALYSIS                0004990
                                        0005000
                                        0005010
                                        0005020
                                        0005030
% CU FLAG WORD                          0005040
                                        0005050
                                        0005060
                                        0005070
                                        0005080
                                        0005090
                                        0005100
                                        0005110
                                        0005120
                                        0005130
                                        0005140
                                        0005150
                                        0005160
                                        0005170
                                        0005180
                                        0005190
                                        0005200
                                        0005210
                                        0005220
```

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Department of Computer Science University of Illinois at Urbana-Champaign Urbana, Illinois 61801	UNCLASSIFIED
	2b. GROUP

REPORT TITLE

WEATHER ANALYSIS ON A PARALLEL COMPUTER

DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research Report

AUTHOR(S) (First name, middle initial, last name)

David Martin Goddard

REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1, 1969	71	14

a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
US AF 30(602)4144	DCS Report No. 365
b. PROJECT NO. 46-26-15-305	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

0 DISTRIBUTION STATEMENT

Qualified requesters may obtain copies from DCS.

1. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Rome Air Development Center Griffiss Air Force Base Rome, New York 13440

3 ABSTRACT

The numerical weather analysis procedures of the National Meteorological Center are reviewed, and problems in adapting them to a parallel computer are considered. Algorithms for storage allocation, processing teletype data, dynamically arranging data, and producing the analyzed field are presented.

DD FORM 1 NOV 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
ILLIAC IV						
parallel computation						
numerical weather prediction						
meteorology						
upper air analysis						

OCT 15 1973

FEB 7 - 1974

UNIVERSITY OF ILLINOIS-URBANA
510.84 IL6R no. C002 no.361-366(1969
Digital computer internal report /
3 0112 088398877